EMPIRICAL ANALYSIS OF EXISTING AND EMERGING THREATS AT SCALE USING DNS

A Thesis
Presented to
The Academic Faculty

by

Charles Lever

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology
May 2018

EMPIRICAL ANALYSIS OF EXISTING AND EMERGING THREATS AT SCALE USING DNS

Approved by:

Assistant Professor Emmanouil Antonakakis, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Douglas Blough
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Mustaque Ahamad
School of Computer Science
Georgia Institute of Technology

Associate Professor Roberto Perdisci
Department of Computer Science
University of Georgia

Professor Fabian Monrose
Department of Computer Science
University of North Carolina at Chapel Hill

Date Approved: December 15, 2017

To my wonderful family, for their endless love and support.

Couldn't have done it without y'all.

ACKNOWLEDGEMENTS

During my time at Georgia Tech, I've been fortunate to work on a plethora of research projects spanning many different security topics—allowing me to work with a lot of very gifted researchers. I am very thankful for this experience as it allowed me to grow as a researcher and build relationships that will last a lifetime.

I'd like to thank both of my advisors (former and current) throughout my PhD journey for taking a chance on me. Patrick Traynor immediately took me on as a student upon entering the program and provided invaluable early direction and support. Not only did I learn how to do research with Patrick, but I made numerous lifelong friends while working in Patrick's lab. I will be forever grateful for my time under his advisement. When faced with a difficult decision of potentially leaving Georgia Tech, Manos Antonakakis offered me a home in his lab and has provided invaluable guidance and mentorship ever since. I cannot thank him enough for taking me on as his student and helping me to become the researcher I am today. Without either of these two individuals, I would not have been able to publish the research upon which this thesis is based.

In addition to my advisors, I have been fortunate to work with many incredibly smart and thoughtful people—many of whom will be lifelong friends. I'd like to thank Brad Reaves and Hank Carter for their support and friendship throughout my time in the PhD program. From my first day on campus, you all truly made Georgia Tech feel like a place I belonged. Additionally, I'd like to thank Yacin Nadji, Panagiotis Kintis, and Yizheng Chen for welcoming me into the Astrolavos Lab family. Each of you made the ups and downs of doing research a little more manageable, and I cannot thank you enough for your assistance and friendship along the way. To everyone else that helped or provided feedback on research, thank you so much and know that you contributed to the work presented in this thesis.

I'd also like to thank my family for all their love and support over the years. Mom and Dad, I would never have had the opportunities necessary to reach this point without your unending and unwavering support. Chapman and Grace, thank you for helping me remember that there is a life outside of the PhD program. To each of you, know that this achievement was made possible because of all the help and support you provided along the way. I couldn't have done it without you, and I love you all.

Last but not least, I'd like to thank my committee members—Fabian Monrose, Mustaque Ahamad, Roberto Perdisci, and Doug Blough. Thank you for all your feedback, suggestions, and guidance throughout this process. I've enjoyed getting to work with you over the years and I feel honored that each of you agreed to sit on my committee.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The security landscape is constantly evolving. Therefore, in order to build better defenses, it is critical to evaluate emerging and existing threats to better understand how and where to prioritize future security efforts. Ideally, such evaluation of threats should be based on real world data, but this introduces a number of challenges. In particular, real world data must be collected, parsed, and cleaned before any sort of analysis can proceed.

The work in this thesis provides an empirical analysis of numerous existing or emerging threats using real world data at scale. As such, it provides the first real world study on the emergence mobile malware by studying network traffic from almost 25M devices—showing that security practices on popular mobile device platforms appear to be fairly effective. In addition, it studies the unintended security consequences of hundreds of millions of domain expirations over several years and shows that malware is increasingly using expired domains for abuse—as well as providing a lightweight algorithm for detecting such expirations. Finally, it studies the evolution of 27M malware collected over almost a half decade—confirming some existing findings at scale and identifying several shortcomings of the current state of the art.

CHAPTER I

INTRODUCTION

From the rise of mobile malware [91] to botnets resulting from insecure IoT devices [45], the types of threats seen on the Internet are constantly evolving as technology progresses. This leaves security researchers and practitioners with a seemingly endless list of threats to defend against—making it challenging to figure out how to prioritize security efforts. To help guide the usage of security resources and build better defenses, the security community should be able to rely on empirical analysis of existing and emerging threats. Ideally, this analysis should be based on real world data, but beyond simply obtaining such data, this introduces a number of challenges. Real world data is inherently noisy and, consequently, requires cleaning before any sort of meaningful analysis can proceed. Thus, studies based on empirical data must often enhance or refine existing data analysis techniques to derive useful insights.

Given the diversity of Internet threats, another challenge is identifying where to collect the data used for analysis of existing and emerging threats. Data collected from host-based sensors can potentially offer very rich, detailed information about threats; however, host based sensors must be deployed on all devices in a study. Clearly, this is untenable for Internet threats that may span a geographically diverse set of networks operated by different organizations—each with hosts that are running different architectures and operating systems. Fortunately, the Internet is based on a common suite of network protocols. At the core of these is the Internet Protocol (IP), which provides the fundamental means of addressing resources on the network. IP addresses can be difficult to remember and may frequently change over the lifetime of a network resource. To address this problem, the Domain Name System (DNS) acts as a phonebook for the Internet, mapping easy to remember domain

names to IP addresses. Both applications and users rely on DNS when accessing Internet resources such as websites and network services. DNS is fundamental to modern IP based communication networks, and given its importance on the Internet, it provides an excellent vantage point from which to perform analysis of Internet threats at scale. Not only is it agnostic of the underlying platform, but data collected from popular recursive DNS servers can provide insights into traffic from a large number of devices. This thesis presents several large scale, empirical studies of existing or emerging threats—each of which rely on DNS for measurements at scale.

The first study discussed in this thesis addresses the emerging threat from mobile malware. With the rise of smartphones and tablets, malware targeting these platforms has grown over time. This has drawn the attention of security researchers who have spent considerable effort trying to characterize mobile malware [80, 81, 91, 252, 253] through a range of static and dynamic analysis techniques. However, unlike traditional computing devices, mobile devices tend to rely on first party software markets for distribution of mobile applications, and each of these markets implements its own security safeguards to try and prevent malicious applications [59, 138]. Beyond software markets, the security policies of mobile applications themselves are enforced differently than on mobile devices—enforcing finer grained permissions of capabilities and using sandboxing for application isolation. Despite these stringent security safeguards, the growth of malware targeting mobile devices seems staggering [87]. However, prior to our study, it was not well understood how widely the mobile ecosystem was actually infected with mobile malware. Thus, a major goal of this study was to better understand infection rates in the mobile ecosystem and to discover if the infrastructure used by mobile malware actually differed from that of traditional malware.

The next study empirically analyzed the growing threat caused by expiring domain names. As previously discussed, DNS is fundamental to IP based networks. Consequently, domain names are often used as anchors of trust, where simple ownership of a domain is enough to attest one's identity. Unfortunately, domain ownership can change and, when

this happens, the trust in that domain is implicitly inherited by the future owner—creating an opportunity for the new owner to abuse that trust. An extremely common form of domain ownership stems from the expiration and subsequent re-registration of a domain name. In our study, we show that this is exceeding common by observing hundreds of millions of domains expiring over a period of seven years. We found instances of expired domains associated with a number of different uses from open source software repositories, browser plugins, redundant services, and e-mail addresses used by critical services. The consequence of an attacker re-registering these domains varied even though the underlying cause remained the same. DNS based blacklists and reputation systems are also affected by this phenomenon as malicious actors can leverage the good reputation of an expired domain when they re-register the domain and repurpose it for abuse. A major goal of this study was to empirically measure the prevalence of this abuse over an extended period of time and present some potential remedies based on our findings.

Our final study presents a longitudinal analysis of nearly thirty million malware samples over a half decade. Many systems have been proposed to statically and dynamically analyze malicious software and produce detailed behavioral reports [169, 244] over the last decade, which has resulted in vast amounts of data collected by such systems. As a result, researchers now have access to more malware related data than ever before, but previous studies [105, 171, 173, 216, 248, 251] often used small datasets and performed narrowly focused analysis—studying topics like the role of cloud providers, the infrastructure behind drive-by downloads, or the domains used by few malware families. Given the wealth of malware data currently available, a major goal of this study was to shed light on how the infrastructure and methods used by Internet miscreants have evolved over time. This enables us to confirm existing results, unearth new behaviors, and identify trends that may require further investigation.

Ultimately, the goal of this thesis is to shed light onto different threats facing the Internet through empirical analysis. Through multiple empirical studies of Internet threats, we

presented three key insights that were previously unknown to the security community. First, the study of mobile malware empirically demonstrated that very few mobile devices appear to be infected with mobile malware—challenging the conventional wisdom of the time—and that mobile threats look similar from the perspective of network infrastructure. Next, we showed that not only is residual trust abuse the underlying cause of many seemingly disparate security problems, but it is also a threat that is increasingly being leveraged by malware—showing consistent growth year over year—in our in our empirical study of domain expirations. Finally, our longitudinal study of nearly 27M malware samples over half a decade demonstrated that malware samples are frequently discovered weeks or months after the threat is visible on the network—suggesting that systems that rely on malware may result in large windows of vulnerability for organizations that rely on them. Not only do our studies provide valuable insights into several existing and emerging Internet threats, but they underscore how DNS provides an excellent platform agnostic data source for studying such threats.

1.1 Contributions

Empirical Analysis of Mobile Malware: The advent of mobile operating systems like Apple's iOS and Google's Android has resulted in mobile devices that are increasingly capable and easy to use. Such devices have seen rapid adoption and, as a result, are increasingly attractive targets for malware authors. To better understand the magnitude of the threat posed to mobile devices, we perform the first empirical study of mobile malware using traffic from a major cellular provider in the United States [150]. Not only does this study allow us to quantify the threat in terms of real subscribers, but it also enables us to analyze if the network communication from mobile threats differs from traditional malware. Lastly, we introduce several enhancements for DNS filtering that were necessary to help distinguish between mobile and non-mobile devices in cellular networks.

Empirical Analysis of Expired Domain Abuse: The domain name system (DNS) is

fundamental to IP based networks, and on the Internet, domains are frequently treated as anchors of trust. Simple ownership of a domain is often enough to gain the trust of users as well as network applications and services. We introduce the notion of *residual trust* in the context of expiring domains, discuss how abuse of residual trust is the root cause of a number of seemingly different security problems, and empirically measure the extent to which residual trust is being abused by malware [152]. Additionally, we develop a lightweight algorithm for locating potential domain ownership changes using only DNS data—providing a possible technical remedy for addressing residual trust abuse.

Empirical Analysis of Traditional Malware: Malware is at the center of many security incidents on the Internet, and understanding how malware evolves and communicates is critical when attempting to build defenses against it. Therefore, we performed a longitudinal study over a half decade, which looks at nearly thirty million malware samples, to understand the evolution of different types of malware and features of their network communication [151]. We observed that potentially unwanted programs (PUP) have become more prevalent than traditional malware in recent years, and PUPs exhibit different network behavior than traditional malware. Furthermore, we provide several insights into how malware commonly uses domain names and the infrastructure those domains resolve to. To facilitate this analysis, we enhance several existing techniques in order to filter and cluster the data used in our analysis.

1.2 Dissertation Overview

We start by providing the background necessary to understand the content presented in this thesis, followed by a discussion of related work, in Section 2.1.

Next, we present our empirical study of mobile malware using network traffic, from a cellular carrier in the United States, in Chapter 3. The methodology used to perform this study, including enhanced techniques, is discussed in Section 3.2. A discussion of the datasets used in this study follows in Section 3.3, and finally, we conclude this chapter with

a discussion of the results in Section 3.4.

We continue with another empirical study of expired domain abuse in Chapter 4. Before presenting the empirical results of our study, we discuss several case studies of residual trust abuse in Section 4.2. Then we proceed with our empirical analysis of residual trust abuse in Section 4.3 followed by presentation of a new technique for detecting possible instances of residual trust abuse in Section 4.4. We conclude this chapter with a discussion of potential remedies for this emerging threat in Section 4.5.

In Chapter 5, we present the results from our longitudinal study of nearly thirty million malware samples over a half decade. We begin with an introduction to the datasets used to perform our study in Section 5.2. This is followed by a discussion of how we performed the filtering and classification of these datasets in Sections 5.3 and 5.4 respectively. We conclude this chapter with a presentation of our findings in Sections 5.5 through 5.8.

Chapter 7 concludes this thesis, starting with a summary of contributions. We then proceed with a discussion of the limitations of our work in Section 7.1. Finally, we leave the reader with closing remarks in Section 7.2.

CHAPTER II

BACKGROUND

2.1 Background

2.1.1 The Domain Name System

The domain name system (DNS) is a backbone protocol for the Internet that maps easy-to-remember domain names to IP network addresses. The domain name space is arranged as a tree, beginning with a root node. In the DNS hierarchy, under the root node are the *top-level domains* (TLDs), and under the TLDs are the *second-level domains*, and so on. Common TLDs include *com.*, *net.*, and *uk.*. A *fully qualified domain name* (FQDN) includes all domain levels that describe the node in the DNS tree; for example, the FQDN *www.example.com.* contains the TLD (*com.*), the 2LD (*example.*), and the 3LD (*www.*). A large portion of the domains are registered directly under a TLD. In some cases, however, this is not possible; therefore, domains under which users can directly register a new domain name are often considered *effective TLDs*. The canonical example is that many domains in the UK are registered under *co.uk.* and not under *uk.* directly.

The basic type of information link in DNS is the *resource record* (RR). DNS defines a number of RR types. For example, an A-type RR links a domain name with an IPv4 network address, while a CNAME-type RR links a domain name with another "canonical" domain name [164, 165].

RRs are returned in response to a DNS query from a requester. Figure 1 illustrates the DNS query process from a host for the A-type record for *example.com.*. A DNS query is initiated by a DNS *resolver* running on a host. This application is responsible for generating some sequence of queries and translating the responses to arrive at the requested resource. There are two parts in a typical DNS resolution request: the recursive and iterative part.

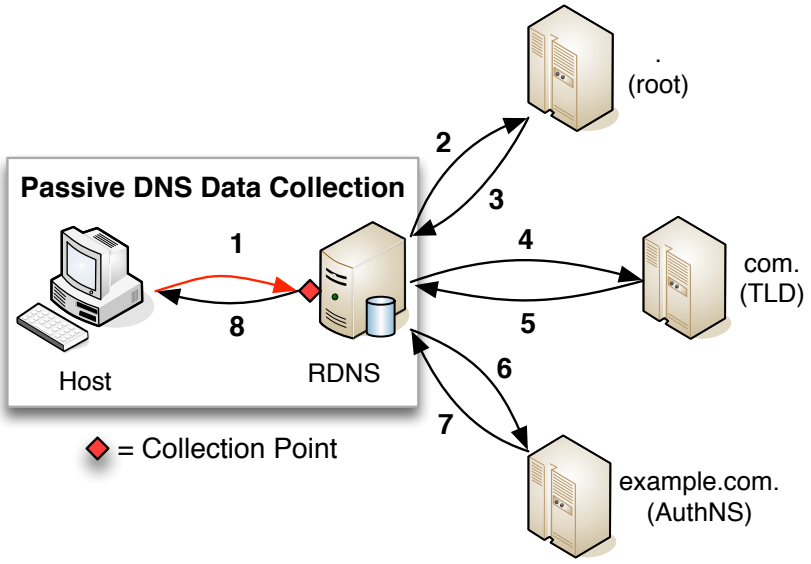


Figure 1: Example of the domain resolution process.

In a typical use, an end system will issue a recursive request using a *stub resolver* to a dedicated *recursive DNS resolver* (RDNS) (Step 1, Figure 1). In a recursive request, the RDNS is charged with completing the iterative portion of the DNS resolution process. It will communicate with the necessary remote name servers (NS) and returns a DNS answer, from the authoritative NS for the requested domain, to the stub resolver in the form of an RR-set. In the case of Figure 1, the RDNS sends iterative requests to the various levels of the DNS hierarchy (Steps 2–7). In Step 7, the RDNS receives the authoritative answer for *example.com.*, and sends it to the requester (or stub resolver) in Step 8, completing the DNS resolution. The RDNS will typically cache the RR locally for up to some period, the Time To Live (TTL), specified in the RR. This improves efficiency and reduces the load on the DNS infrastructure.

2.1.2 Passive DNS Monitoring

Since a RDNS mediates all requests from a client's stub resolver, it is possible to perform passive DNS (pDNS) data collection of DNS queries received at the RDNS. This pDNS data collection typically includes all of the information associated with the successfully resolved

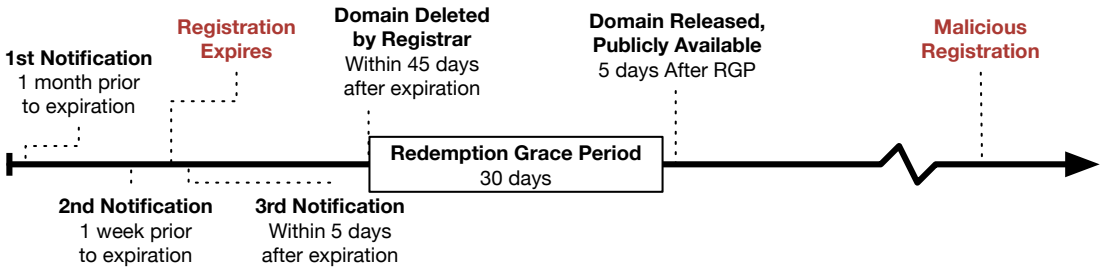


Figure 2: Timeline of a domain expiration.

DNS queries by the RDNS.

There are several benefits to using pDNS monitoring for DNS analysis. Malicious queries are able to be logged and analyzed without alerting the owners of the malicious domains (unlike DNS probing [118, 157]). Another benefit of pDNS data collection is that it can allow the discovery of malicious domains not previously known to exist on DNS blacklists (DNSBL) [43, 44, 61] and does not require previous knowledge of the domain's existence. A potential drawback of pDNS data collection at the local RDNS level is that the data collected will be limited by the amount of traffic handled by the RDNS. Therefore, it is important to collect a large number of queries from a geographically diverse set of RDNS servers.

2.1.3 Domain Registration Process

We define the term *residual trust* as the historical reputation of a domain that is implicitly transferred with changes in ownership. In this section, we detail the process governing a domain's expiration. In the following sections, we explain how these expired domains can be exploited by abusing the domain's residual trust.

Domain names are registered, owned, and expired using processes created by Internet Corporation for Assigned Names and Numbers (ICANN) in conjunction with registry operators and registrars. With a few exceptions, domains are typically registered for a period of one or more years, after which the registrant (i.e., owner) has the option to renew.

As a domain registration approaches its expiration date, it begins the formal ICANN expiration process. For generic top-level domains (such as .com, .net, and .info) the expiration process is governed by ICANN's Expired Registration Recovery Policy (ERRP) [125]. We summarize this process in Figure 2 and discuss the details below.

ICANN's expiration process is intended to address several past and potential abuses such as "domain sniping", whereby a vigilant "domainer" would *register* the domain seconds after expiration and extort a price to transfer the domain back to the former owner. Under the current process, domainers hoping to speculate on expired and lapsed domains must now wait until the release event, giving the current registrant time to renew the registration even after the domain expires.

Specifically, the ERRP requires registrars attempt to notify the lapsed owners (twice prior to expiration, once after). However, in practice, many owners cannot be reached due to a variety of reasons including inaccurate registration information, general neglect, or "tucked" domains. The latter reason, tucked domains, refers to situations where the contact information for the domain resides entirely under the expiring DNS zone itself. For instance, the registrar contact information, WHOIS information, and start of authority SOA RNAME [164] may be entirely under the expiring zone.

After the domain expires, the registrar will delete the domain from the TLD zone causing it to enter a 30-day Redemption Grace Period (RGP). Typically, deletion occurs within 1–45 days after expiration, but the exact length of time may vary due to extenuating circumstances or provisions in the myriad registrar and registry agreements. While in the grace period, the expired domain may still be renewed by the previous registrant, but this is typically at a higher cost. The domain is released five days following the conclusion of the RGP and becomes available for re-registration by others.

There are other variations of the domain expiration process. For example, the Canadian Internet Registration Association uses a "To Be Released" (TBR) process where expiring domains are listed along with all homonyms. For example, `cardreaders.ca` is TBR

listed along with all accented variations such as `çardreaders.ca`, `cárdreaders.ca`, and other permutations. The 30-day process includes a short advance bid auction followed by general release.

Since many expiring domains are valuable brands, large groups of "drop-catchers" pool their resources to attempt registration in the first seconds after release. In order to prevent DDoS-style events against the registries, many providers stagger the release of expiring domains and publish the specific hour (and often the specific minute) during which a given domain will become available. Since valuable dropped domains are generally acquired within seconds, this strategy minimizes the period over which large volumes of registration attempts are directed against the registry.

Despite the post-expiration deletion phase, during which the domain is typically unreachable, third party users will often still attempt to connect to the domain. Increasingly, these connections are through automated tools, and users are often unaware the domain is even absent from DNS. For example, operating systems may attempt to update installed packages through an automated (e.g., cron, launchd) process. Browser plugins may contact home sites upon application startup. Software sharing tools may create connections to numerous file sharing sites on startup in order to obtain timely updates and routing tables stored in distributed hash tables. All of the domains associated with these automated activities can and do expire. Therefore, the party acquiring the expired domain has thousands and even millions of users contacting the site. We discuss specific examples and the security implications of this phenomenon in the next section.

2.2 Previous Work

2.2.1 Malware Infrastructure

A number of works have studied the malicious infrastructure used to distribute malware. For example, Rossow et al. [198] performed a large scale analysis of malware downloaders and their network infrastructure. Using the analysis traces of samples belonging to 23

downloader families, the authors discovered that 20% of the C&C servers remain operable long-term. In particular, the authors identified 2,942 C&C domains, which resolved to 861 IP addresses hosted in a variety of different ASs. Moreover, they found that malware infrastructures regularly migrate among different domains, while keeping a redundant presence in several different providers. In our experiments we are not able to distinguish C&C from other forms of malware traffic. However, in Section VI we discuss several characteristics of the hosting infrastructure of the domains resolved by the samples in our dataset and we found that some families use the same set of IP addresses for very long periods of time. In particular, PUP seem to have extremely stable infrastructures.

Wang et al. [234] built honeyclients to find drive-by download websites that exploit browser vulnerabilities. Similarly, Moshchuk et al. [168] used honeyclients to crawl over 18 million URLs, finding that 5.9% contained drive-by downloads. Provos et al. [188] studied the prevalence of drive-by downloads and the redirection chains leading to them, finding that 67% of the malware distribution servers were hosted in China [187]. These papers shed light on other aspects of the malware infrastructure, for instance by pointing out that malicious files are often hosted on multiple servers reachable by many different URLs at the same time. While malware samples can also occasionally contact infected web pages (e.g., to download additional components) our study focuses on the domains contacted by malware after a system has been infected and not on their drive-by download infrastructure. In a study closer to our work, Polychronakis et al. looked at the network behavior of malware distributed by drive-by downloads [183] using light-weight protocol responders. However, the authors focused on the purpose of the network activity (e.g., reporting home, data exfiltration, or joining a botnet) more than on the infrastructure and domains used by the malware authors. Active probing techniques have been proposed to measure the size of the server infrastructure for specific malware families [175, 246].

Another relevant line of work studies rogue networks and autonomous systems hosting unusually large amounts of malicious activity. To this end, Fire [211] used approximately

one year of data from the Anubis system and Shue et al. [206] a dataset collected over a period of one month.

More recently, researchers investigated the use of cloud hosting providers as infrastructure for malware. These studies were conducted either by performing some form of active probing [174, 233] or by mining the information extracted from dynamic analysis sandboxes [111]. In this last case, the authors analyzed over 1M malware samples which connected to at least one publicly routable address on the Amazon EC2 Cloud. While this work did study network infrastructure, the results were limited to the Amazon cloud and required a considerable amount of manual analysis to separate and classify the different types of communication. By using our larger datasets we were able to confirm this trend and observe a similar effect on a more global scale, affecting multiple cloud providers - in particular from 2014. We also found that PUP families are the ones that rely the most on this type of stable infrastructure.

2.2.2 PUP Infrastructure

Recent work has studied the prevalence of PUP [142, 220], its distribution through pay-per-install (PPI) services [141, 222], and its detection [144, 145]. Thomas et al. [220] showed that ad-injectors affect 5% of unique daily IP addresses accessing Google. They also measured that Google's Safe Browsing generates three times as many detections for PUP as for malware [222]. Kotzias et al. measured that 54% of 3.9 M hosts they examined had PUP installed [141] and that PUP dominates so-called malware feeds [142]. Kwon et al. detect PUP and malware distribution using graph-based approaches leveraging machine learning [144] and temporal properties [145]. While some of these works explore PUP-related domains, none of them analyze properties of the PUP domain and server infrastructure. Most related are the categorization of the top 15,000 pages driving traffic to PPI downloaders by Thomas et al. [222] and the analysis by Kotzias et al. [141] of the top 20 e2LDs from where PUP is downloaded. Compared to our work, those analyses cover

different time periods, only a fraction of PUP domains, and more importantly do not explore infrastructure properties such as PUP domain lifetime and hosting. In summary, we believe we are first to analyze the properties of the PUP domain and server infrastructure.

2.2.3 Longitudinal Malware Studies

While our study focuses exclusively on the network infrastructure, other researchers investigated the behavior extracted from dynamic analysis sandboxes to study other characteristics of malware samples.

One of the first attempts in this direction was performed by Bayer et al. [58] using almost 1M samples collected until 2009 by the Anubis platform. Interestingly, in this early study the authors reported that only 47.3% of the samples that showed some network activity also performed a DNS query—which succeeded in over 90% of the cases. Lindorfer et al. [154] performed a similar experiment focusing on the Android malware landscape. In this case, the authors reported that 99.91% of Android malware performed DNS queries, with roughly one third failing to resolve—suggesting an increasing adoption of domain generation algorithms (DGAs). While these studies provided interesting data points, they were performed at a distance of five years on datasets over 25 times smaller than the one used in this paper.

Several other papers have analyzed the use of DGAs, which can be used by botnets to bypass domain blacklists. Kolbitsch et al. [140] used binary code reuse [64] techniques to extract the DGA of the Conficker.A botnet and use it to compute the set of domains used on a given date. A related reverse-engineering approach was used by Plohmann et al. [182] to perform a large-scale analysis of DGAs used in malware. A different approach to detect and analyze DGAs is Pleiades [50], which monitors unsuccessful DNS resolution requests from recursive DNS servers in large networks. Our experiments confirm the wide use of DGA, and find that up to two thirds (67%) of the fully qualified domain names queried by malware failed to resolve to a valid address.

2.2.4 Internet Reputation

Network level analysis of malicious behavior offers a complementary means of characterizing and mitigating malware. For example, a popular method of preventing or limiting the spread of malware is the use of Internet blacklists. IP blacklists provide a list of known bad actors in the form of IP addresses which network operators can subsequently block; however, the use of DNS to build malicious network infrastructure has grown due to its resilience against IP blacklisting [199, 205].

Consequently, a significant amount of work has focused on analyzing network abuse at the DNS level [74, 112, 114, 136, 155, 237]. This has led to the creation of systems that are able to detect malicious domains through the use of passive DNS monitoring and machine learning [43, 44, 62]. Ultimately, these systems allow network operators to assemble DNS blacklists of malicious and suspicious domains in order to detect and prevent malicious activity on the network.

This has led researchers to study the usefulness of such blacklists. Metcalf et al [160] performed a comparison of 86 different internet blacklists—of both varying category and type—over a span of 30 months. Unfortunately, the inventory of blacklists appears to be partially anonymized, and some blacklists were collected for as little as three months. Therefore, it is difficult to compare these results directly with the blacklists used in this work. The major findings from their work showed that there is little overlap between lists for fully qualified domain names or IP addresses and, when there is overlap, no lists consistently outperforms another. Kührer et al. evaluated the effectiveness of 19 malware blacklists [143] collected over 2 years by classifying the entries (i.e.,non-existent, parked, or sinkholed), measuring the completeness, and testing the reaction time of each blacklist. When compared against a dataset of 300K samples, they show that the union of all blacklists contains 70% of domains detected per family. Furthermore, 58% of domains were seen in their passive DNS an average of 334 days before appearing on a blacklist. Our study shows similar results over a much longer observation period and with almost 100 times more malware.

Finally, Rajab et al. proposed a content-agnostic malware protection system based on binary and IP/DNS reputation to address the shortcomings of both blacklists and whitelists [191], and other works have been proposed that leverage network related reputation data for malware detection [134, 189].

2.2.5 Expired Domains

There has been a wealth of research focused on using DNS as a tool for detecting malicious behavior. For example, researchers have previously used elements of DNS to classify malicious websites [66, 158]. Other researchers have used DNS information to understand and predict future malicious behavior [89, 115, 194, 200] and identify previously unknown malicious domains [48, 49, 51, 185, 190, 247]. In addition to using DNS for prediction and detection of malicious infrastructure, other work has focused on protecting the domain name system itself from abuse [54, 76]. Even commercial entities frequently use DNS-based tools to help protect against known malicious domains through the use of blacklists [208].

Our understanding of expired domain abuse first came from early research into the fate of failed banking domains by Moore and Clayton [167]. Their study focused on expired financial sites and found some instances where old, failed bank web sites were re-registered and likely used for nefarious purposes. However, the study authors were narrowly focused on methods for detecting failed banking domains.

Unlike this previous work, we study how *residual trust*—implicitly transferred between owners of a domain name—affects the security of systems and entities that rely on DNS. Our multi-year study demonstrates that residual trust abuse is being actively exploited and the problem is growing. Further, our work shows that this phenomenon impacts prior work by the security community and, thereby, demonstrates the need for more research into residual trust and malicious re-registrations.

2.2.6 Mobile Malware

The importance of mobile networks is increasing as society becomes more reliant on mobile devices such as smartphones, tablets, and mobile broadband cards. Several works have examined mobile device network traffic to learn about the general network characteristics of those devices [85, 88, 101]. Past studies have shown that certain design considerations have made these networks inherently vulnerable to Denial of Service (DoS) attacks. Traynor et al. [226, 227] proposed a text messaging DoS with only the bandwidth of a cable modem. This research demonstrated a growing class of vulnerabilities due to the increasingly intertwined connectivity between the Internet and traditional voice networks. Other work has shown that the use of data communication protocols on voice networks creates the potential for failure under modest loads [197, 228–230]. Accordingly, significant effort has now been directed towards the analysis of potentially malicious mobile applications.

Numerous studies have highlighted the weaknesses and potential for misuse of various aspects of the Android security model [70, 83, 90, 92, 93]. Other work on Android devices suggests that it is difficult to tell if an application breaks any phone-wide security policies [82] and has resulted in tools to aid in the analysis of Android applications [80, 81]. Additional studies have surveyed the types of malware seen in the wild and evaluated the efficacy of different techniques in preventing and identifying such threats in the future [91]. However, app analysis alone provides an incomplete picture of the current state of malware on mobile devices and networks.

Network level analysis of malicious behavior offers a complementary means of characterizing and mitigating malware. For example, a popular method of preventing or limiting the spread of malware is the use of Internet blacklists. IP blacklists provide a list of known bad actors in the form of IP addresses which network operators can subsequently block; however, the use of DNS to build malicious network infrastructures has grown due to its resilience against IP blacklisting [199, 205]. Consequently, a significant amount of work has focused on analyzing those networks at the DNS level [74, 112, 136, 155, 237]. This has

led to the creation of systems that are able to detect malicious domains through the use of passive DNS monitoring and machine learning [43,61]. Furthermore, recent work has shown that detection of malicious domains can also be accomplished by passively monitoring DNS at the upper levels of the DNS hierarchy; this allows DNS operators to independently detect malicious domains without relying on local recursive DNS servers [44]. Ultimately, these systems allow network operators to assemble DNS blacklists of malicious and suspicious domains in order to detect and prevent malicious activity on the network.

Though there has been considerable effort targeted towards detecting network malware, it has been focused primarily on traditional wired networks. The question of whether such threats differ or even exist in real mobile networks has yet to be evaluated through empirical results.

CHAPTER III

EMPIRICAL ANALYSIS OF MOBILE MALWARE

3.1 Motivation

Malware writers have begun to pay attention to mobile phones. In response, a significant amount of effort has been spent by researchers to characterize malware in mobile applications markets [80, 81, 91, 252, 253]. These efforts have applied a range of static and dynamic analysis techniques on a large number of applications in an attempt to discover malicious code. Market operators including Google and Apple have also invested significant resources in an attempt to prevent malicious applications from being installed on mobile devices and for later removing such applications if necessary. However, for all of these efforts, the extent to which the mobile ecosystem is actually infected by such malware is not well understood. Without such an analysis, it is impossible to determine whether or not current defense mechanisms are having any demonstrable effect.

In this chapter, we discuss the results of the first network level analysis of mobile malware using traffic from a major cellular network. We work from the hypothesis that malicious mobile applications are not different from the bots and malware in the non-cellular world in that they rely on the same core functionality of the Internet in order to achieve scale and robustness. In particular, we began our research with the belief that malicious behavior in the mobile environment similarly relies on the same Internet hosting infrastructure used to support traditional malware activities including propagation and update (e,g., a malware download site), command and control (e.g., communication with infected devices), and data transfer (e.g., a site to upload of stolen data). Should this hypothesis prove to be true, more traditional network-based techniques for detecting and combatting malware [43] can potentially be applied in this new space. As research on botnets and malware have shown,

such an Internet-based approach is more effective and scalable than malware-analysis based approaches, particularly in the face of (future) mobile malware with stealth, obfuscated logic, such as triggered-based behaviors.

3.1.1 Contributions

We verify our hypothesis experimentally using three weeks of DNS data from a major US-based cellular provider collected over the course of three months. We first show that the vast majority of the hosts resolved in the cellular dataset are also seen in a separate DNS dataset from a non-cellular ISP. After this confirmation, we dig more deeply into the cellular dataset and uncover a number of important results regarding malicious behavior in cellular networks, including the following contributions:

First, we observe that known mobile malware samples are virtually unseen. We extract DNS domains from large public and private datasets of mobile malware and specifically search for their resolution in our dataset. Our analysis demonstrates that only a vanishingly small number of mobile devices appear to be infected: 3,492 out of 380,537,128 devices, or less than 0.0009% of the population. This lends credence to the argument that while the mechanisms market operators implement to protect users from malware may be bypassable [59, 138], malware writers are failing to infect mobile devices with much success. Like any application developer, a malware writer faces the challenging task of developing a popular application (or a malicious application in disguise) that will be downloaded by a large population. That is, the probability that a user will download an unknown malicious application is very small. In addition, a legitimate application market will remove any known malicious application, further reducing the probability of a mobile malware being downloaded.

Next, we compare traffic to suspicious hosts from iOS against all other devices. Common opinion argues that the closed nature of Apple's App Store and operating system make

devices in this ecosystem more secure. However, our analysis demonstrates that approximately 8% of iOS-based devices communicate with known suspicious hosts, virtually the identical frequency as all other mobile platforms. Accordingly, users of these devices do not appear to be any more or less likely to communicate with potentially malicious hosting infrastructure than other users. To say the least, iOS does not appear to provide any more effective mechanisms to prevent users from engaging in unsafe activities.

Lastly, we observe campaigns with mobile malware clients by obtaining traffic from the upper layer of the DNS hierarchy and analyzing two major threats with mobile malware clients. We see that the lifetime of these two threats lasts on the order of months, and that the criminal operators make use of network agility and move their hosting infrastructure over time (e.g., changing domain names and IPs). Finally, we also note that one of these campaigns cease operating long before the mobile malware associated with each campaign is ever identified, lending credence to the possibility that network-based countermeasures may help identify and mitigate (e.g., via DNS blacklisting) such threats faster than the analysis of mobile malware.

3.2 Methodology

In this paper, we analyze DNS data generated by devices subscribed to a major US cellular carrier. For simplicity, we use the term "mobile devices" to describe smartphone and tablet platforms (e.g. Android, iOS, and others). Our primary focus is Internet-based *hosts* contacted by mobile devices using DNS. By *host* we mean a remote IP address included in a successful *A-type* DNS domain name resolution. We focus on the hosts as opposed to domain names explicitly because of the extensive forensic evidence already connected with malicious behavior in the Internet (e.g., C&C, drive-by, PPI, etc.). We link the hosts observed in DNS resolutions within a cellular network with historic evidence of Internet abuse and reveal the extent to which mobile devices contact hosts historically associated with known malicious behavior.

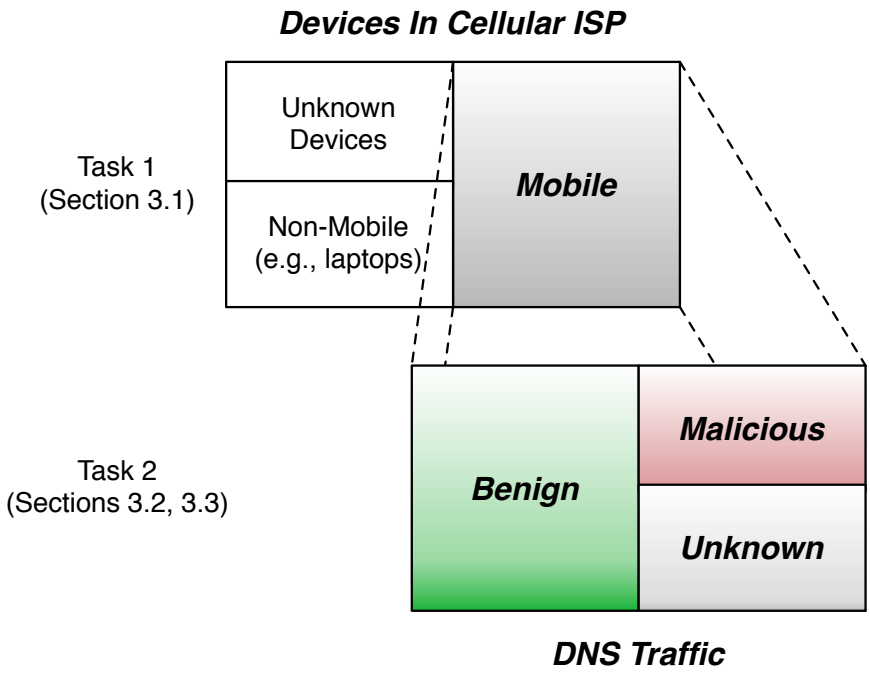


Figure 3: A high-level view of the two tasks required to identify malicious behavior by mobile devices. First, all traffic generated by non-mobile devices is filtered out of our dataset. Second, the remaining traffic is characterized as either benign, malicious or of unknown reputation.

Our monitoring point is a single sensor that aggregates traffic collected from several RDNSs providing DNS resolution services for mobile devices located across many different US states. We perform two high-level tasks in order to characterize malicious behavior generated by mobile devices. First, we remove traffic generated by non-mobile devices (i.e., laptop and desktop machines also serviced by this provider via hotspots and "cellular connect cards") and then attribute each request to a specific mobile device; second, we perform a reputation analysis of the RRs associated with these mobile requests and classify traffic as having benign, malicious or unknown reputation. This workflow is shown in Figure 3, and the details of each task are provided in the remainder of this section.

We use the following notation to more formally describe our workflow. A resource record (RR) refers to either an *A-type* or *CNAME-type* record and its corresponding RDATA [164, 165] tuple of *(qname, ip)* and *(qname, domain)* respectively. We represent a DNS query, q_j, for a domain name, d, and the related DNS response, r_j, as the tuple $\mathcal{Q}_j(d) = (T_j, R_j, \boldsymbol{IPs}_j)$,

where T_j identifies the epoch in which the query-response was observed, R_j is the unique identification of the device that initiated the query q_j, d is the queried domain, and $\boldsymbol{IPs}_j$ is the set of resolved hosts as reported in the response r_j.

3.2.1 Mobile Device Identification Process

Accurately identifying individual devices is challenging due to IP address churn and device roaming in a cellular carrier. In order to ensure that we do not include non-mobile devices in our evaluation, we must first be able to attribute traffic to specific devices. We received proprietary data from the carrier that allows us to definitively attribute traffic to a mobile device operating on the network for each epoch; however, we cannot correlate devices across epochs.

We formally define the set of mobile devices in an epoch. Let $MDEV_i = \{R_k\}_{k=1...l}$ be a set that contains all l mobile device identifiers (R_k) for epoch i, and $i \in EPOCHS$ [1]. We then define the set of domains resolved by $MDEV_i$ in epoch i as $mDN_i = \{d_k\}_{k=1...n}$, where n is the number of domains (d_k).

Mobile devices are restricted to certain ranges of device IDs that are only associated with operation over a cellular data connection (i.e., 3G or 4G). While this excludes most traditional computing devices, traffic generated by a laptop or desktop using cellular connect cards or tethering through a mobile phone remains in our dataset. We remove this traffic by examining the resource records associated with each device and looking for mobile-specific domains. While there is no official standard, there are some common indicators for mobile-specific domains. One is the use of mobile-specific subdomains such as *m.example.com* or *mobile.example.com*. Another method is the use of the URI path as a mobile indicator, but since we are working with DNS data, we do not have access to this type of indicator. Table 1 provides a subset of mobile indicators.

In addition to mobile indicators, some domains are strongly or exclusively associated

[1] We define the set of epochs as *single days* falling within the following ranges $EPOCHS = \{4/15/12 - 4/21/12, 5/13/12 - 5/19/12, 6/17/12 - 6/23/12\}$

with mobile applications. Such sites include mobile ad networks, mobile application programming interfaces, and mobile services. For example, Google's AdMob [30] advertising network is only supported on Android and iOS, so devices that contact the AdMob network are almost certainly mobile. Certain mobile APIs use unique domains that are easily identifiable. Apple's Push Notifications, for instance, use a set of mobile domains (e.g., `*.courier.push.apple.com`) reserved specifically for the push notifications service. Additionally, services like HeyTell [2] provide push-to-talk functionality for mobile devices via a mobile application. Devices communicating with these types of services should be almost exclusively mobile.

We first exclude all devices R_k that query any of six domain names d in an epoch j related to standard operations of Windows operating systems [2]. Devices are labeled as mobile only if they contact a domain d with a mobile indicator or domains strongly associated with mobile applications or services. We are able to identify $132,516$ unique domain names that fit this former category.

Through the combination of filtering based on device IDs and mobile domain inference, we are able to identify devices as either mobile or non-mobile with high confidence. In the small number of cases where overlap exists, we tag a device as unknown and do not consider its behavior as reflective of mobile devices. By conservatively choosing devices in this manner, we strongly reduce the likelihood of selecting traditional computing devices that are connected via mobile broadband cards or tethering. However, if a traditional computing device is visiting a mobile resource, we would falsely label that device as mobile. We note that this scenario is exceedingly rare as browsers generally direct users to the appropriate version of a website.

[2]Two of the most frequently hit domain names in this list are the time.windows.com and download.windowsupdate.com.

Table 1: Examples of popular mobile indicators.

Domain name	**Indicator Type**
m.example.com	Subdomain
mobile.example.com	Subdomain
android.example.com	Subdomain
iphone.example.com	Subdomain
ipad.example.com	Subdomain
touch.example.com	Subdomain

3.2.2 Filtering Benign Domains

At this point in our workflow, we can label a device and associate it with RRs. We can then begin the second task in Figure 3: classifying each of the resource records requested by mobile devices. The first step in this process is identifying and removing known benign traffic from our dataset. Our methodology again remains **conservative**, aggressively reducing false positives potentially at the cost of increasing false negatives. We achieve this by whitelisting all requests made to the top 750,000 effective second level domains (e2LDs) according to Alexa [40]. However, we must note that we do not whitelist domains associated with dynamic DNS (DDNS) providers given their common use by network malware. Intuitively, such broad whitelisting removes the most popular sites (and e2LDs) as they are more likely to be trustworthy and less likely to be intentionally malicious. This approach is commonly used in DNS-based reputation and classification systems [43, 44].

We want to remain as conservative as possible and reduce any potential false positives from our datatset. To that end, we further filter benign traffic from our dataset by removing a number of the most popular remaining e2LDs. We compile a list of approximately 800 e2LDs based on the lookup volume of the queried domains. We manually inspect all of them and we classify them as benign. We should note that the lookup volume distribution for the e2LDS follows a power law.

The end result of the whitelist filtering process is a reduced set of domain names $mDrdc_j = \{d_k\}_{k=1...n}$, where $mDrdc_j$ is a set that contains all n **not whitelisted** domain

Algorithm 1: The algorithm used to obtain the set of IP addresses (or hosts) that represent the hosting infrastructure that facilitated resolution of domain names from mobile devices in epoch j.

INPUT : $MDEV_j$, $mDrdc_j$ and all $\mathcal{Q}_j(d_i)$ for every mobile domain name ($d_i \in mDrdc_j$) observed in epoch j.

Let $HOSTS_j = \emptyset$, be the set that will contain the unique hosts (or IP addresses) that have been mapped with the domain names in $mDrdc_j$ after the completion of the process.

[1] : $\forall d \in mDrdc_j$:

[2] : Let IPs_j be the set of IPs in the tuple $\mathcal{Q}_j(d)$, if $R_j \in MDEV_j$

[3] : $HOSTS_j \cup IPs_j$

OUTPUT : $HOSTS_j$

names (d_k) resolved by mobile devices in epoch j. In this set we will have domain names in the "Malicious" and "Unknown" categories of Figure 3.

3.2.3 Feature Extraction

The remaining entries in our dataset now belong to mobile devices communicating with Internet-based hosts with either malicious or unknown reputations. We now describe the features that we extract from these remaining domains, which will allow us to analyze the **hosting infrastructure** supporting these domains. We use Algorithm 1 to find all unique IP addresses (hosts) for all domains in the set $mDrdc_j$ for epoch j, resulting in the set $HOSTS_j = \{IP_k\}_{k=1...n}$. For ease of understanding, the entire feature extraction process is summarized in Figure 4.

We compare the traffic observed in the cellular carrier (only from the mobile devices) against a pDNS data collection from a non-cellular ISP. Let $f_{pdns}(d) = \{ip_k\}_{k=1...n}$ be a mapping function that takes a domain name d as input and returns a set of routable IP addresses that have been historically linked with d.

The $f^j_{cell}()$ function returns passive DNS data from the DNS traffic in the cellular carrier over an epoch j. Let $f^j_{cell}(ip) = \{d_k\}_{k=1...n}$ be a function that receives an IP address ip as

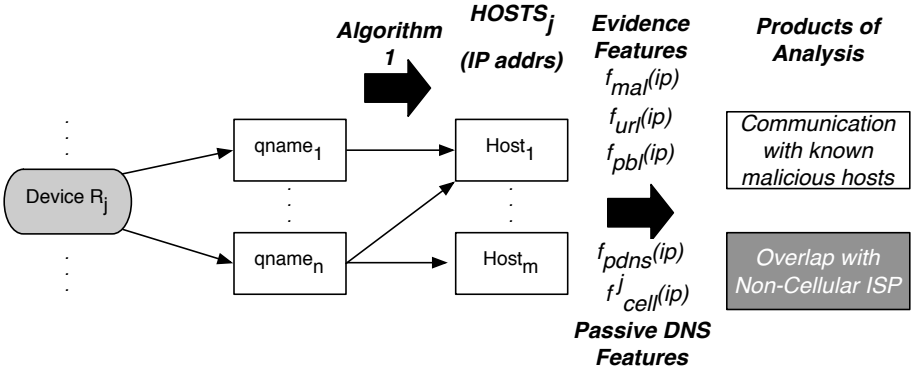


Figure 4: Determining the communication patterns for each mobile device (R_j). Each qname requested by R_j is converted into an IP address via Algorithm 1. This list of IP addresses ($HOSTS_j$) is then processed for Passive DNS Features (PF) to determine overlap with traffic from our non-cellular ISP and for Evidence Features (EF) to determine the presence of communications with known malicious domains.

input and returns a set of related historic domain names (d_k) observed in the cellular network during epoch j from mobile devices in set $MDEV_j$.

The function $f_{mal}(ip) = \alpha$ returns the number α of unique malware samples that IP address ip has been associated with over the past 19 months. The association could be direct (i.e., the malware contacts the IP address/host) or indirect (i.e., the malware looks up a domain name that resolves to that IP address/host) as shown in Figure 4. We similarly define $f_{url}(ip) = \beta$ and $f_{pbl}(ip) = \gamma$, as the functions that return the number of malicious URLs (β) and malicious entries (γ) in public blacklists.

For every host (ip) in the set $HOSTS_j$ we extract the two groups of features: passive DNS and evidence features. *At the end of the feature extraction process, we obtain statistical historic passive DNS and malicious evidence-based observations (used in Section 3.4.1) for the set of hosts in $HOSTS_j$.*

Passive DNS Features (PF) We collect two features from this group. They are simply the number of elements in the sets $f_{pdns}(ip)$ (i.e., related historic non-cellular domains) and $f^j_{cell}(ip)$ (i.e., related historic cellular domains) for an address ip.

Evidence Features (EF) We compute a total of three features from this group. These features describe direct reputations of the IP addresses in the set $HOSTS_j$ (during the

epoch j). We compute three features for each IP address: (i) $f_{mal}(ip)$, the count of unique malware associated with ip, (ii) $f_{url}(ip)$, the count of URLs associated with ip, and (iii) $f_{pbl}(ip)$, the count of public blacklisting incidents associated with ip.

Both PF and EF feature families represent the basic building blocks of DNS reputation systems [43, 61]. We select them to understand (i) the extent that malicious hosts currently serve mobile-related DNS resolutions and (ii) the extent that the infrastructure used to resolve mobile-related domain names is already present in passive DNS data collections from non-cellular networks. In particular, the PF feature family, which is based on passive DNS data, will show to what extent the hosts from mobile RRs (directly or indirectly) can be associated with DNS resolutions from a non-cellular ISP. The EF features, which are based on historic reputation information, will show us to what extent the already tainted Internet hosting infrastructure is currently used directly by mobile devices. Additionally, we perform one more level of filtering in which we evaluate the malicious hosts identified by the EF feature family using the Notos [43] reputation system; we remove any hosts identified using the EF features if Notos does not produce a reputation score below our chosen threshold.

3.3 Dataset Summary

This section describes the datasets used in our analysis. These include pDNS data collected from a major US cellular carrier, pDNS data collected from a major US non-cellular ISP, and a database of malicious evidence built from several classes of malicious information.

3.3.1 DNS

Cellular

We observed DNS traffic from a cellular data network on twenty-one days over a three month period. This data was passively collected from a single sensor that aggregates information from several cities.

Table 2: Listing of unique RRs, domains, hosts, and devices seen in cellular dataset.

	Duration (hours)	**RRs**		**Domains**	
		Total	*New*	*Total*	*New*
4/15-4/21	168	8,553,155	8,553,155	8,040,141	8,040,141
5/13-5/19	168	9,240,372	4,498,765	8,711,704	4,042,009
6/17-6/23	168	8,660,555	3,246,194	8,109,536	2,745,999
Total	504	26,454,082	16,298,114	24,861,381	14,828,149

	Duration (hours)	**Hosts**		**Devices**	
		Total	*New*	*Total*	*Mobile*
4/15-4/21	168	2,070,189	2,070,189	157,286,931	121,497,066
5/13-5/19	168	2,168,266	606,467	169,561,760	136,292,358
6/17-6/23	168	2,050,168	377,048	153,525,716	122,747,704
Total	504	6,288,623	3,053,704	480,374,407	380,537,128

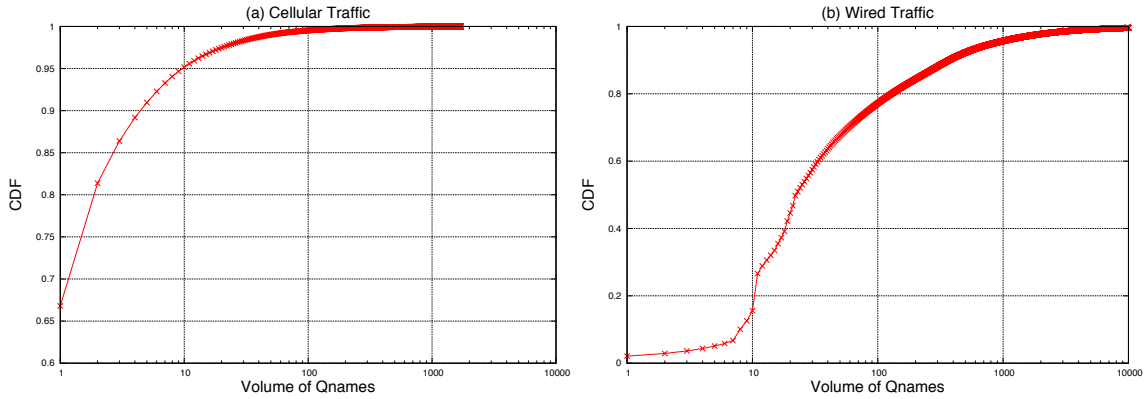


Figure 5: Per host (in $HOSTS_{all}$) qname distribution of DNS evidence. These experiments demonstrates that the hosts in $HOSTS_{all}$ have a significant historic presence in pDNS data collected from non-cellular networks.

3.3.1.1 Observations from the Cellular Carrier Traffic

Table 2 provides insight into the number of *unique* RRs, domains, and hosts seen over the twenty-one (single day) epochs. For each record type, there are two columns that specify the total number of unique records seen for the given week and the number of new records not seen in any prior week. Intuitively, the number of new records seen should decrease over time, and Table 2 shows that the influx of RRs, domains, and hosts does follow this pattern.

Non-cellular

The non-cellular pDNS data was collected from seven different sensors located across

the US over more than 15 months. Due to the extended collection period, this dataset presents a substantial volume of traffic that can be used to provide historical context for domains and hosts of interest. In particular, we can use this data to make inferences about the hosting infrastructure of a particular domain or tie specific hosts to their related domains.

3.3.2 Devices

Devices seen in the cellular dataset accessed the network via a cellular data connection. Consequently, these devices should fall into three general categories: smartphones, tablets, or mobile broadband devices. The first two categories include devices such as Android and iOS phones and tablets. A mobile broadband device includes any device accessing the network via a mobile broadband card or tethering to another device's cellular data connection. This could include traditional computing devices such as desktop or laptop computers.

As discussed in Section 3.2.1, we are conservative in the classification of mobile devices; a comparison of the total devices and what was classified as a mobile device can be seen in Table 2. Most importantly, this table shows that our estimate of mobile devices is conservative; we classify only 79% of devices seen as mobile.

3.3.3 Evidence

We analyze cellular DNS traffic with an evidence database composed of three general classes of non-mobile malicious evidence: public blacklist data (PBL), phishing and drive-by download evidence (URL), and hosts accessed by known malicious applications (MAL). In addition to the non-mobile evidence database, we also used a mobile blacklist (MBL) containing 2,914 domains known to be associated with mobile malware or mobile malware operators. Figure 6 shows the volume of DNS lookup requests from mobile devices in our cellular dataset that could be directly linked with different classes of non-mobile and mobile evidence.

Additionally, a diurnal analysis of the volume of requests that can be directly associated

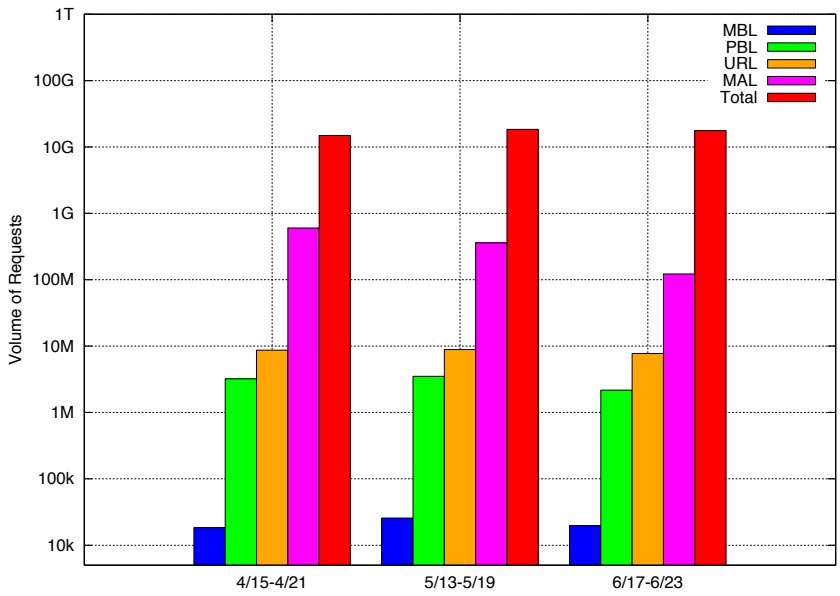


Figure 6: Volume of requests to domains with malicious evidence visited by mobile devices in cellular network.

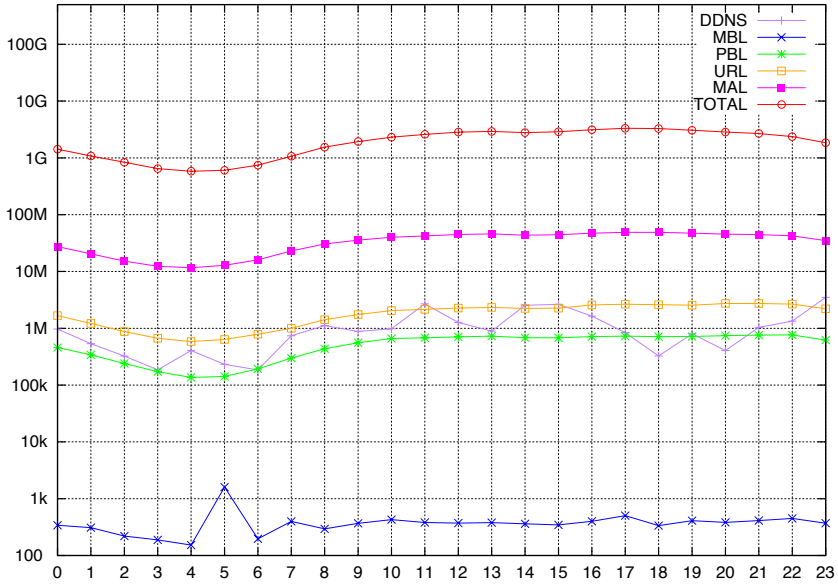


Figure 7: Hourly analysis of request volume for various types of domains observed from mobile devices.

with different types of malicious evidence can be seen in Figure 7. It is important note that our cellular DNS traffic sensor aggregates data from several different locations in different time zones. Therefore, patterns may be less pronounced than if all data was collected from a single location. Looking at the total volume of requests for each hour, Figure 7 shows that

the volume of traffic starts to increase between hours five and six and begins to gradually decrease between hours seventeen and eighteen. These hours approximately correspond to people waking up in the morning and traditional dinner hours in the evening. With the exception of DDNS traffic and a spike at hour five for MBL requests, the other classes of domains appear to follow the total hourly volume pattern.

3.4 Results

We now present the results of our experimental evaluation. We begin by analyzing the traffic observed in the cellular carrier, first characterizing the data from the cellular carrier in isolation, and then comparing the request patterns to those observed historically in our non-cellular ISP dataset. We then examine the extent to which the hosts observed in mobile resolution requests are directly or indirectly tainted by reputation information collected in non-cellular ISPs.

We continue in Section 3.4.2 and further focus on evidence of mobile-specific malware. We continue our analysis of directly tainted hosts in mobile resolutions by determining from which mobile platforms those queries originate. Then, we examine all of the cellular network queries for mobile-malware specific domains to determine the extent of the presence of mobile malware in the cellular network.

We conclude our results with long term analysis of two known mobile threats in Section 3.4.3. In particular, we study those two threats from their rise until they become almost completely inactive. Furthermore, we provide the global infection perspective of those two mobile threats. Finally, we examine their hosting infrastructure and how it changes over time.

3.4.1 Analysis of the Reputation Datasets

We analyze the DNS traffic generated from approximately 380 million mobile IDs over the course of our observation period (Table 2). Note that while we can identify devices consistently within an epoch (i.e., a single day), we can not link devices across days, meaning

that this total does not represent the number of unique devices serviced by the carrier. We are specifically interested in unique Internet-based hosts requested by mobile devices. The filtering process results in the set $HOSTS_{all}$ consisting of 2,902,071 unique hosts having the following characteristics: (i) at least one resolution request for each host was observed by the cellular carrier, (ii) the observed DNS requests came strictly from devices classified as mobile, (iii) the hosts are not associated with any known benign infrastructure, and (iv) the host is a routable IP address. We obtain passive DNS information on each of these hosts from our historical non-cellular ISP dataset (via the function $f_{pdns}()$ defined in Section 3.2.3).

3.4.1.1 Observations from the Cellular Carrier Traffic

We use the hosts in $HOSTS_{all}$ to perform an in depth examination of their DNS properties in our pDNS data. We do this for both the cellular ISP traffic (Figure 5(a)) and, using a projection into our passive DNS data collection, for the non-cellular ISP traffic (Figure 5(b)).

The hosts in $HOSTS_{all}$ reflect a portion of the hosting infrastructure that support unknown or malicious types of DNS resolutions in the mobile carrier. We project this set into the non-cellular data collection and obtain non-cellular passive DNS data for the hosts in $HOSTS_{all}$. Only 36,338 (or 1.3%) of hosts in $HOSTS_{all}$ are outside the non-cellular passive DNS evidence we have.

Looking a bit closer, Figure 5(a) shows the distribution of unique hosts in the set $HOSTS_{all}$. We see that more than 18% of the hosts requested by mobile devices are associated with only a single domain. Furthermore, Figure 5(b) shows that 98.7% of hosts in the set $HOSTS_{all}$ have at least one historically associated domain name (according to the passive DNS data collection from the non-cellular ISP). This simply means a sufficiently large non-cellular pDNS data collection can be used to amplify the DNS information for hosts observed in DNS resolutions from non-cellular networks.

If we assign Notos [43] reputation scores to the RRs that have IPs in the $HOSTS_{all}$ set, and use reputation threshold of 0.8 (or above 80% probability of the RR being malicious),

Table 3: M&A Datasets

Market/Dataset Name	Market Country	Date of Snapshots	#Unique Apps	#Unique domains	#Unique IPs
Google Play*	US	9/20/11 and 1/20/12	26,332	27,581	47,144
SoftAndroid	RU	2/7/12	3,626	3,028	8,868
ProAndroid	CN	2/2/12, 3/11/12	2,407	2,712	8,458
Anzhi	CN	1/31/12	28,760	11,719	24,032
Ndoo	CN	10/25/12, 2/3/12, 3/6/12	7,914	5,939	14,174
Contagio	—	3/27/12	338	246	2,324
Zhou et al.	—	2/2012	596	281	2,413
M1	—	3/26/2012	1,485	839	5,540

*Top 500 free applications per category only

we obtain 51,503 domain names (3,636 distinct IPs) as likely suspicious. Only 18 domain names (13 unique hosts) of them have been listed in mobile black lists until the day of the submission. While 0.8 is a conservative threshold, these results could be used as an indicator that the malicious hosting infrastructure observed in cellular networks is already present in reputation and passive DNS observations from non-cellular networks.

These findings are already valuable. Given the significant overlap between the domains requested by devices in cellular and non-cellular providers, and the historical information regarding the reputation of the hosts in $HOSTS_{all}$ (as discussed in this Section and Section 3.3.3), we can conclude that the DNS infrastructure (malicious or not) is being reused in cellular networks. Moreover, the scores assigned to hosts by DNS reputation systems can potentially serve as filtering functions for applications when they are submitted to mobile markets.

3.4.1.2 Observations from Mobile Markets and Malware Datasets

We now characterize application markets and datasets of known mobile malicious applications. Specifically, we examine all of the applications in Proandroid, Sofandroid, Anzhi, Ndoo and the top 500 free applications from each category in Google Play. We also analyze three datasets containing known mobile malware, including all of the malware samples from

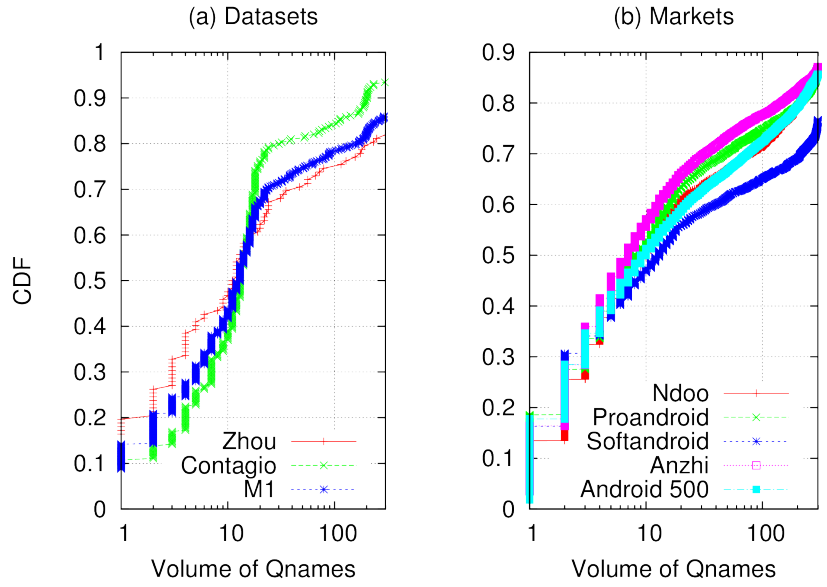


Figure 8: Qname distribution per passive DNS (from non-Cellular networks) evidence gathered from the $m\&a$ dataset. In spite of their geographic diversity, the requested qnames in all of these subsets follow a similar distribution.

the Contagio blog, 596 apps from the malware dataset described in Zhou et al. [252] and a third dataset which we refer to as $M1$ that was provided confidentially by an independent security company. $M1$ contains both highly suspicious and confirmed malicious applications. We refer to this collection of datasets as the malware and applications ($m\&a$) datasets ($\{Android500, Proandroid, Softandroid, Anzhi, Zhou, Contagio, M1\}$). We statically extract domain names from application code in the $m\&a$ dataset to create a set of domain names we call $DN_{m\&a}$. Using the $f_{pdns}()$ function, we record the unique hosts historically associated with each entry in $DN_{m\&a}$ from our non-cellular pDNS dataset. Table 3 shows the breakdown of each of these datasets.

Figures 8(a) and (b) show the distribution of qnames for each of the subsets of the $m\&a$ dataset. We observe that at least 90% of the hosts in $HOSTS_{m\&a}$ are present in our non-cellular pDNS dataset, and in some cases (see Figure 8(b)) all of them are present. Despite the geographical diversity of the markets we examine, in Figures 8(a) and (b) we see that all $m\&a$ datasets have very similar distributions of qnames per host in $HOSTS_{m\&a}$. Furthermore, more than 50% of these hosts have at least seven domain names historically

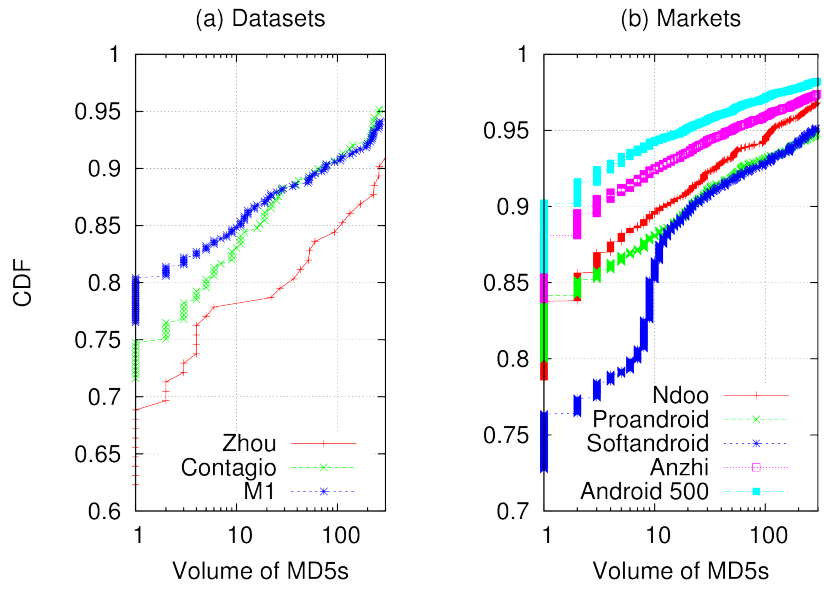


Figure 9: Volume of malicious MD5 evidence associated with unique qnames seen in the $m\&a$ dataset.

associated with the (non-cellular) pDNS dataset.

Figure 9 shows the direct relationship to hosts historically associated with malware for each of the subsets of the M&A datasets. We observe that the 38% of the hosts retrieved from the $HOSTS_{Zhou}$dataset, which corresponds to applications from Zhou et al. [252] dataset, have been historically associated with more than one malware samples (or MD5). Additionally, approximately 23% of hosts in $HOSTS_{softandroid}$ dataset, can also be linked with more that one malware samples.

Somewhat unsurprisingly, Google Play (see Figure 9(b), Android 500 class) has the lowest percentage of applications contacting hosts historically associated with malware. Only a 10% of the domain names observed in the class Android 500 in Figure 9(b), have more than one malware sample associated with the host the domain names resolved to historically. These numbers increase in Figure 10, which shows the projection of $HOSTS_{m\&a}$ through our passive DNS data collection from non-cellular networks using $f_{mal}()$. As previously mentioned, such indirect results are prone to false positives due to phenomena like parking IPs and sinkholes; however, the inclusion of such hosts in an application could easily serve

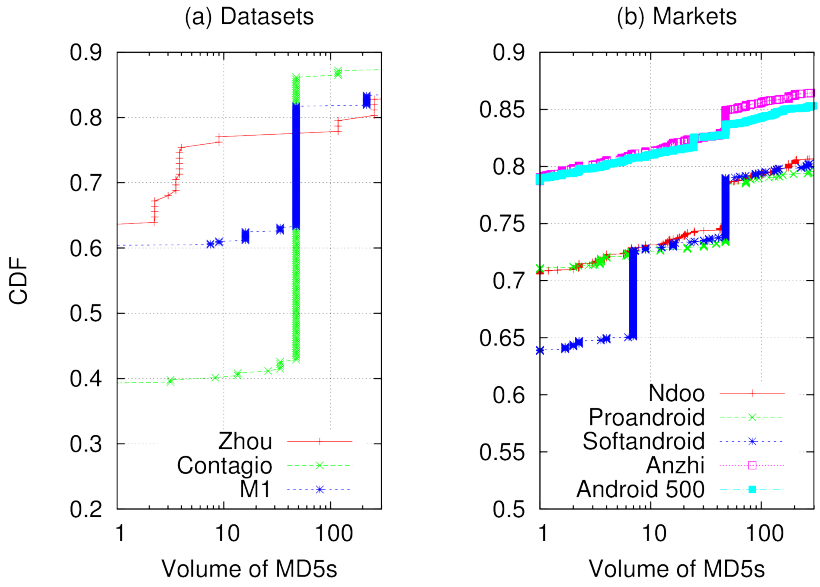


Figure 10: Volume of malicious MD5 evidence associated with unique qnames in the $m\&a$ dataset after projection through non-cellular pDNS data collection.

to trigger analysis by market operators.

In summary, mobile applications observed from a variety of marketplaces and malware datasets use much of the same Internet infrastructure as the non-cellular DNS resolutions. This observation is similar to the analysis we made in Section 3.4.1.1, when we examined DNS traffic from a large cellular provider. Perhaps the most important observation from the $m\&a$ datasets is that some market-places (e.g., Softandroid) contain mobile applications that tend to be more directly and indirectly related with known malware-tainted hosting infrastructure.

3.4.2 Mobile-malicious activity in mobile networks

This subsection discusses two distinct but related phenomenon: 1) the relationship between mobile platforms and requests for tainted hosts, and 2) the presence of queries for domains facilitating malware that targets mobile platforms.

3.4.2.1 Tainted hosts requested by mobile platforms

Table 4 presents a breakdown of which platforms correspond to what proportion of total mobile device population, what proportion of the device population requested tainted hosts, and which platforms are responsible for tainted host requests. This data is presented for iOS devices and other mobile devices such as Android and other indistinguishable platforms (e.g., potentially Symbian, Windows Mobile, unverifiable iOS devices). We separate iOS devices from the rest because they are easy to reliably identify by searching for push notification domains.

The first column in Table 4 shows the contribution of each platform to the total population of mobile devices. The majority of devices were unidentifiable, but roughly one third of the devices could be identified as iOS. The second column shows the percentage of devices for each platform that requested domains that point to tainted hosts. We observed roughly 8% of iOS and 8% of other mobile devices issued at least one request that pointed to a tainted host.

Finally, the third column shows how platforms contribute to the overall number of tainted host requests. It is interesting that each class of devices contributed proportionally to the number of total tainted host requests. This data shows that, from the network perspective, iOS devices reach out to similar numbers of tainted hosts as other devices.

Figure 5 provides an overview of different threat classes that are present in our PBL evidence set. Each of the different classes shown in this table were seen in our cellular pDNS dataset and was visited by a mobile device. It is interesting that this list includes several prominent desktop malware classes such as Zeus and SpyEye in our filtered mobile traffic.

3.4.2.2 Mobile malware in the cellular network

To answer the question ”How prevalent is the threat of malicious mobile applications in the cellular network?”, we scanned all (not mobile-device only) cellular network DNS

Table 4: Tainted Hosts and Platforms

Device Platform	% Total requests by mobile device	% Population requesting tainted hosts	% Total tainted host requests
iOS	31.6%	8.8%	33.2%
All other mobile (Android, etc.)	68.4%	8.2%	66.8%

Table 5: Threat Class for Tainted Hosts

Threat Class	# Associated Hosts
Artro	1
Backdoor.Tofsee	1
DMSSpammer	2
FakeAV	1
MalwareDomainList	1386
MalwarePatrol	203
Misused	69
Phishing	383
SC	4
SpyEye	183
Worm.Palevo	30
Zeus	1083

data with a blacklist of 2,932 domains known to be associated with mobile-malware or mobile-malware operators. We then focused on those domains that are known to have directly facilitated mobile malicious activity, not merely associated with it. Examples of this malicious activity include distributing malicious applications, exfiltrating sensitive data without user consent, and command and control services.

We focused on 19 unique domains present in our cellular pDNS data. These malicious domains are associated with 10 unique malware families; all of these are Android applications. 9 of these 10 malware families were publicly disclosed *before* any of our epochs — meaning they were still queried after they were known to be malicious by security researchers, antivirus companies, and market providers. [7–9, 68, 176, 181, 215, 224, 249]

Table 6 shows the mobile malware families with domains seen in the cellular network, the number of domains known to facilitate the malicious activities of those families, the

number of devices of any type and the number of mobile devices that contacted a domain facilitating mobile malware. Note that this data is aggregate across days and all domains facilitating mobile malware; we cannot identify the same device across epochs, so this is an upper bound of devices that contact a domain. The effects of our conservative mobile classification process are apparent – only a fraction of devices that we classify as mobile contacted any domain that facilitates mobile malware compared to all devices.

The most prevalent malicious family in the network was FakeDoc. This is a potentially unwanted application (adware) that steals a user's Google account and other potentially sensitive information. FakeDoc was discovered in the Android market on October 19, 2011 well before our traffic epochs. Despite being flagged by several antivirus products [8], 5,417 devices contacted the domain used by FakeDoc for malicious activity.

The second most popular malware family was NotCompatible. NotCompatible is a trojan application that acts as an open network proxy. Unlike the other families in Table 6, NotCompatible is spread through compromised web pages with hidden iFrames that point to a download site for the app. NotCompatible was disclosed on May 2, 2012 [249].

Even considering an upper bound, the overall traffic to domains associated with mobile malware is quite small. Only 9,033 devices of any type out of a total of over 480M million (0.001%), and 3,492 devices out of a total of 380M confirmed mobile devices (0.0009%) contacted a domain known to facilitate mobile malware. The top two threats present in our data present a stark contrast in functionality, time of known activity, and method of distribution. Despite these differences, neither presented significant activity levels during our measurement epochs.

A number of insights can be gleaned from this data. First and foremost, mobile malware is a real threat to users in the United States, despite the fact that malware researchers find many of their samples in non-US markets. Even though the threat is real, it is minimal. It is important to note that the overall size of all infected populations indicates that mobile malware is far from reaching the scope or severity of desktop malware. This may be

Table 6: Malicious Apps with Domains in Mobile Network

Malware Family	# Assoc. Domains	#Devices (Any type)	#Devices (Mobile only)
DroidDreamLight*†	3	150	44
DroidKungFu*	1	19	6
FakeDoc*†	1	5417	2145
Fatakr*	1	328	151
GGTracker*	3	1	1
Gone60*†	1	1	1
NotCompatible	3	2198	762
Plankton*†	4	686	286
Malware β*	1	18	1
WalkInWat*	1	215	95

* Disclosed before any of our epochs
† Distributed in Google Play market

attributed to moderated markets, security architectures of mobile platforms, and the relative lack of opportunity an infected device can provide a malware author.

The low volumes of traffic from malware distributed through the Google Market indicate that market-based kill switches can be effective at controlling the malware population. However, the *relative* success of NotCompatible calls into question whether the market-based kill switches will be able to control the spread of malware in the future if malware authors eschew markets in favor of other distribution means. Even when markets are used as a distribution channel, mobile malware can be seen in the network long after discovery by researchers and its removal from markets. This finding implies that neither markets nor security products like antivirus tools are able to guarantee a malware-free platform.

3.4.3 Lifecycle of Mobile Threats

In this section we examine the evolution of two malicious mobile applications, threat ϵ and threat β. With historic data from a large authoritative DNS server, we provide insights on how the threats developed over time as well as geographical properties of the requesters requesting domains associated with these threats. We conclude by providing some insight

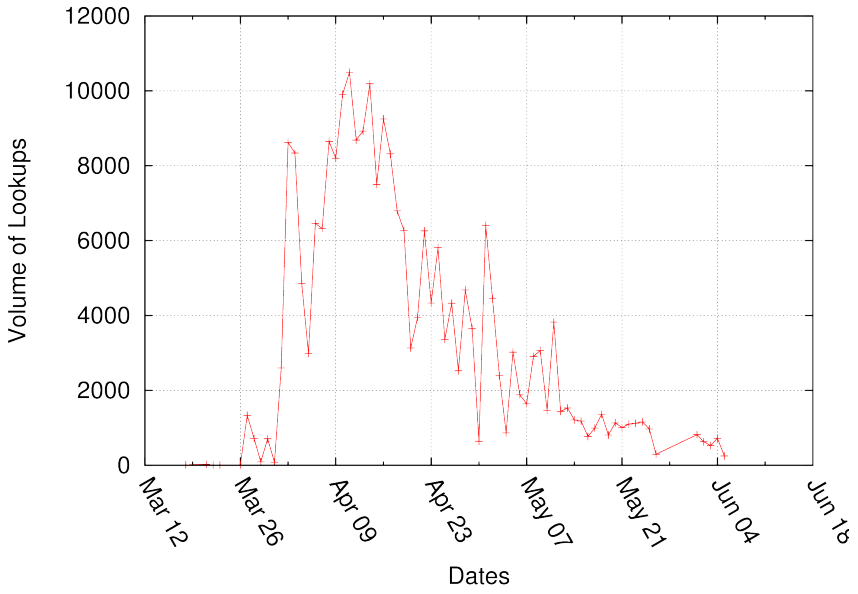


Figure 11: DNS request volume for threat ϵ (2011)

on the hosting infrastructure these mobile threats used throughout their lifetime, especially at the peak of their activity.

Threat ϵ is an Android application that masquerades as a legitimate client to a popular Internet streaming media service. When run, the application presents a credible login screen. When the user attempts to log in, the application displays an error message and closes. In the meantime, it has sent the user's credentials to domain $qname_\epsilon$ in an HTTP request. This threat was publicly disclosed by a major anti-virus company in October 2011.

Threat β is an Android application that starts a service after reboot that periodically contacts a C&C server hosted on domain $qname_\beta$. The service will respond to commands received from the C&C or via SMS. One command causes the application to sign all contacts up to an on-line mailing list, while another command has the application send infected download links to all contacts via SMS. These links are on a different domain than $qname_\beta$. The application will automatically respond to received SMS with an offensive message, and in certain cases will send offensive SMS messages to all contacts. This threat was publicly disclosed by a major anti-virus company in May 2011.

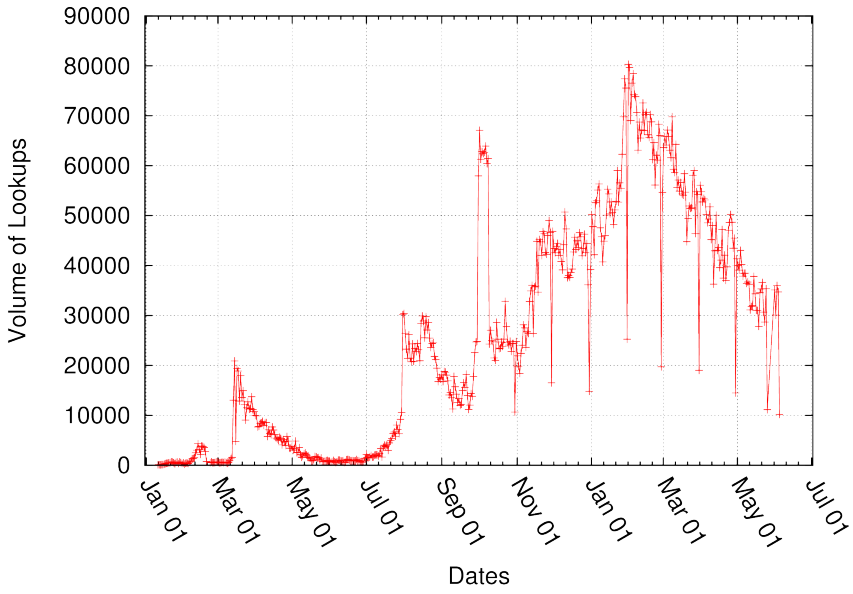


Figure 12: DNS request volume for threat β (2010 to 2011)

3.4.3.1 Lifetime and Infection Scale

Figure 11 shows the daily lookup volumes for $qname_\epsilon$, which acts as a proxy for the victims of threat ϵ. These lookups could be recursive DNS servers, so we cannot make any claims about the size of the overall infected population. The threat was most active on April 12^{th}, but soon after rapidly declines. The first lookup for $qname_\epsilon$ was recorded on March 3^{rd}, 2011, and by June 5^{th} [3] there were DNS requests from 2,731 unique requesters. Table 7 shows the query volume, AS, and country code of the top ten networks that sent requests to $qname_\epsilon$; the majority of these are based in the US. Of note is that this threat seems to have ended well before it was publicly disclosed in October 2011; at the time of disclosure, $qname_\epsilon$ no longer resolved to a routable address.

Figure 12 shows the lifetime of threat β in terms of query volume. This threat became active in January of 2010, and at its peak in February – March 2011 it averaged more than 70,000 DNS requests per day. Over the 14 months that this threat was active, 13,094 unique IP addresses queried the domain name $qname_\beta$. As before, this number cannot be

[3] We have no data from the authoritative DNS server after this date, so we have no visibility into later activity

Table 7: Requester information with respect to autonomous system (AS), country code (CC), and count of unique IPs in the AS (volume).

Threat ϵ			Threat β		
Volume	AS	CC	Volume	AS	CC
816	3356	US	7315	3356	US
112	15169	US	470	3462	TW
97	7132	US	266	15169	US
92	9299	PH	222	4766	KR
67	7843	US	210	7132	US
52	20115	US	160	9299	PH
47	6389	US	139	6389	US
44	7643	VN	127	9121	TR
38	22773	US	122	20115	US
33	24560	IN	115	24560	IN

considered an absolute population estimate. Table 7 shows the distribution of the infected populations for mobile threats β and ϵ. We see that a significant portion of the infected population resides in Asia-based networks. We also note that Google (AS 15169) has a heavy impact on the numbers in Table 7 (most likely due to crawling). Threat ϵ was disclosed well past its peak in DNS requests.

3.4.3.2 Hosting Infrastructure

Here we describe in detail the Internet infrastructure used by both threats. Table 8 shows the autonomous systems, country codes, and the number of hosts within each AS that $qname_\epsilon$ or $qname_\beta$ [4] pointed to throughout their lifetime. Figure 13 shows how the host pointed to by $qname_\epsilon$ changed over time. The host was primarily located in AS 6389, but for brief periods of time the domain resolves to hosts outside AS 6389. Comparing the activity of threat ϵ (Figure 11) to the changes in the host (Figure 13) reveal that host changes were correlated with activity peaks (as seen in April 2011). Figure 14, shows host changes overtime for threat β. Like threat ϵ, the host infrastructure was relatively stable until peak

[4]Only the top four ASes are presented; these comprise all hosts for threat ϵ and 308 out of the 316 hosts used by threat β.

Table 8: Information on the hosting infrastructure used by the two mobile threats.

Threat ϵ			Threat β		
Volume	AS	CC	Volume	AS	CC
11	6389	US	237	6389	US
3	20115	US	28	49544	NL
1	7132	US	28	27589	US
1	13674	US	15	29550	GB

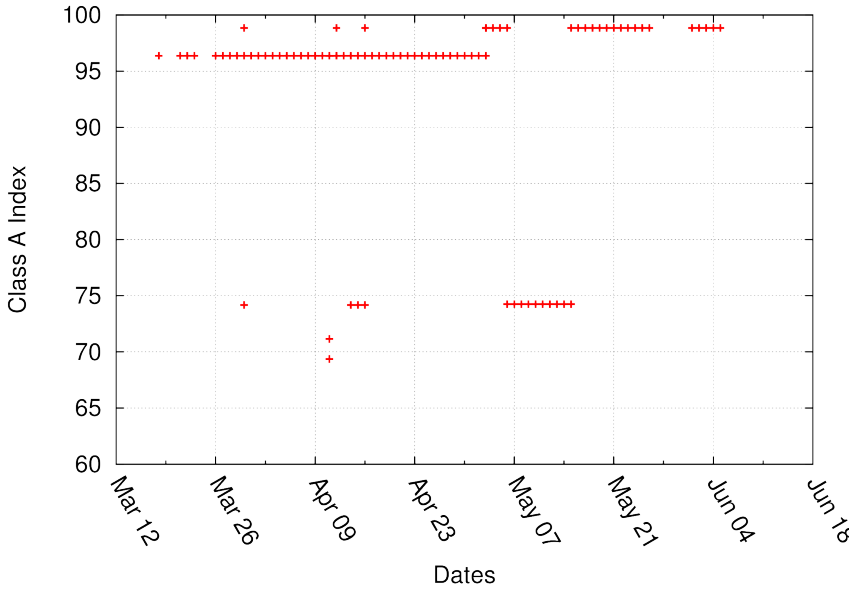


Figure 13: Threat ϵ's host infrastructure shows agility comparable to non-mobile botnets

activity (January – June 2011). The changes in this case may have been to add redundancy to the C&C infrastructure as it grew.

These case studies provide three key insights into the life-cycle of mobile threats. First, in the case of threat ϵ, the malicious app was not publicly disclosed until months after its peak activity. In this case, reactive security measures failed to detect a threat until well after it was most effective. Second, both of these threats show a growth pattern similar to those shown in non-mobile malware studies [44]. Third, the agility in the hosting infrastructure used by these threats does not resemble professional DNS hosting. Rather, they are similar to non-mobile botnet operators that commonly use tactics like moving to hosts in different networks and countries to provide agility to their illicit operations. In future work, the agility

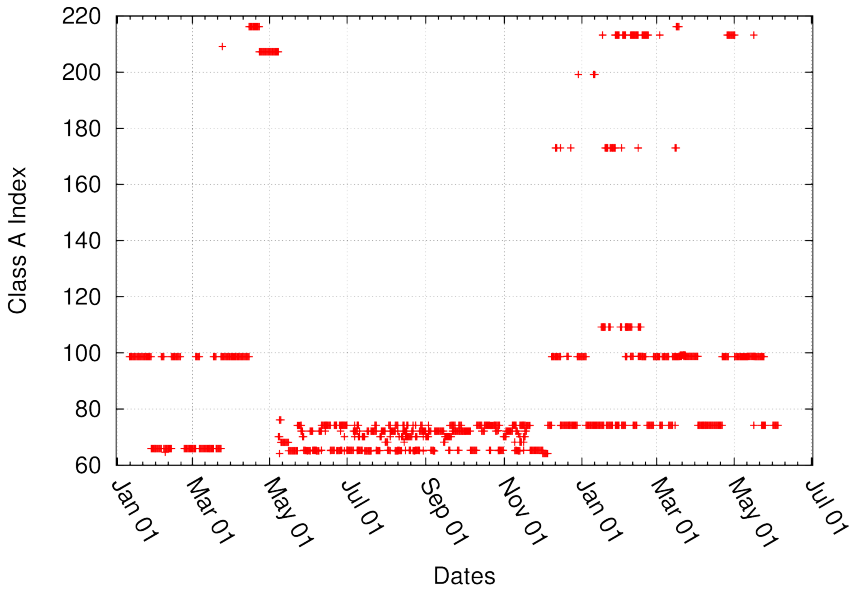


Figure 14: Threat β's host infrastructure also shows agility comparable to non-mobile botnets

seen in these mobile threats may be exploited by traditional DNS reputation systems to detect potentially suspicious domain names in the mobile space.

3.5 Summary

In this paper, we presented a study of traffic obtained from a major US cellular provider as well as a major US non-cellular Internet service provider. Our work provides an in-depth understanding of the Internet infrastructure used for mobile malware. In particular, we showed that the network infrastructure used by mobile applications is part of the core Internet infrastructure used by applications in the non-cellular world; in other words, the mobile web is part of the Internet. We presented evidence showing that the mobile malware discovered by the research community appears in a minuscule number of devices in the network; this suggests that mobile application markets are already providing adequate security for a majority of mobile devices. We compared traffic to suspicious hosts between different mobile device platforms and demonstrated that iOS devices are no less likely than other platforms to reach out to such devices. Finally, we analyzed two major mobile threats

and found that their network characteristics are similar to those of non-cellular botnets. Overall, these findings suggest that there are commonalities, in terms of both network infrastructure and characteristics, between malicious mobile applications and non-cellular malware. Therefore, we should leverage our successful experiences with DNS monitoring and reputation systems for non-cellular ISPs to develop a similar system for cellular carriers to identify (emerging) mobile threats. We leave this as a future work.

CHAPTER IV

EMPIRICAL ANALYSIS OF EXPIRED DOMAIN ABUSE

4.1 Motivation

Domain names have become the Internet's de facto root of trust. In practice, they are also a root of insecurity as common security systems depend on the unfounded assumption that domain ownership remains constant; this leaves users vulnerable to exploitation when domain ownership changes. For instance, authentication systems often rely on email to reset user passwords. Such schemes fail when the domain for that credential changes ownership—e.g., by expiration, auction, or transfer—and thus is no longer associated with the original owner. Consequently, an adversary can exploit this vulnerability to hijack the email address via a malicious re-registration of the domain.

In this chapter, we study the exploitation of domain ownership changes and find that the phenomenon of *residual trust abuse* is the underlying cause of many, seemingly disparate, security issues. Among these, we found vulnerabilities allowing an attacker to maliciously register a domain to: *(i)* siphon University traffic and email by exploiting expired nameserver domains; *(ii)* hijack Regional Internet Registry (RIR) accounts and allocate IP addresses using expired email domains; and *(iii)* distribute malicious updates for benign software, including an instance that left users of a major Linux distribution vulnerable. The preceding examples demonstrate that even a single instance of residual trust abuse has major implications for the security of users and systems alike.

Despite the serious consequences of malicious registrations, the scope of the phenomenon has yet to be rigorously characterized and quantified. Our study seeks to fill this knowledge gap. Using data collected over six years, we show that *adversaries are actively exploiting residual trust*. To quantify this, we analyze the overlap between expired domains and both

(i) hand-curated lists of malicious domains, i.e., public blacklists; and *(ii)* domains queried by malware, as such queries are an indicator of abuse. We find that almost 8.7% of the domain names that appeared on public blacklists (since 2009) were listed after the domains expired and changed ownership. In other words, over the last six years at least 27,758 were abusing residual trust. Similarly, we identified 238,279 domains that expired, were re-registered, and then contacted by malware—indicating likely malicious registrations. These domains account for 3.9% of all domains resolved by malware in our dataset. To put this into perspective, the size of this set is comparable to the 320,009 domains listed on public blacklists since 2009. Even more, empirical *evidence suggests this is a rapidly growing problem*. We found the exploitation of ownership changes has grown by orders of magnitude since we began collecting data. Between 2009 and 2012 there were 784 observed blacklist instances of abuse, but in 2014 alone, that number increased to over 9,000. We observed similar growth for expired domains resolved by malware, indicating this trend is not unique to blacklists.

In light of the increasing abuse of residual trust—e.g., malicious re-registration of domain names—better tools and policies are necessary to ensure the security of both users and systems. We argue that a comprehensive solution must consider both technical and non-technical remedies. For the former we propose Alembic, a lightweight algorithm that can be used to identify likely changes in ownership. This algorithm scales to large amounts of traffic, requires only access to historical DNS data, and ranks likely changes in domain ownership. Using our algorithm, we were able to identify several cases of potential residual trust abuse, including a currently expired advanced persistent threat (APT) domain. The expired APT domain example demonstrates how easily domains with negative residual trust can be used to revive existing infections. For the non-technical remedies, we discuss several potential policy changes and their implementation challenges.

4.1.1 Contributions

We introduce the concept of residual trust and, using numerous real world cases of domain misuse, demonstrate how it is the underlying cause of many seemingly disparate security problems. Furthermore, we distinguish between positive and negative residual trust and discuss how each could be abused or cause unintended consequences.

We provide the first large-scale analysis of residual trust abuse by using several large datasets for expired domains, passive DNS, network malware traces, and aggregated public blacklists. Our observations show malicious parties are actively abusing residual trust and that it is a growing problem.

We propose a technical remedy and discuss several non-technical remedies to help deal with the growing abuse of residual trust. For the former, we introduce a lightweight algorithm, *Alembic*, to help locate likely ownership changes. While identifying changes in domain ownership would appear to be straightforward using WHOIS information [77], mining WHOIS is a challenging and resource-intensive task. Some researchers are trying to solve this problem with better automated solutions [156], but this does not address the problem that simply obtaining WHOIS information is expensive and hard to scale. Further, WHOIS information is rarely available in bulk. It is common for registry access to be limited to just a handful of queries (less than 1000) per day from a given host. While there are commercial companies offering limited API-based access to WHOIS information [17,28,29], they are cost-prohibitive and lack external validation. Due to the previously mentioned WHOIS limitations, it is outside the capabilities of most practitioners, research groups, and all but a handful of organizations to generate a comprehensive set of historic WHOIS records through which domain ownership changes can be identified. These above constraints make building a traditional detection system for domain ownership changes extremely difficult. Therefore, we chose to create an efficient and highly scalable algorithm that helps find *potential* domain ownership changes using only DNS information. Using our algorithm, we find several previously unidentified instances of abuse, including an expired APT domain.

4.2 Abusing Residual Trust

In this section, we discuss five real world examples of residual trust abuse that exploit expired domains previously used for a variety of Internet functions and services—including university DNS servers, CIDR allocations from Regional Internet Registries (RIRs), browser extensions, open source software, and promotional media content. These case studies demonstrate the unintended consequences that result from the residual trust placed upon domains by both users and systems. Our goal is to introduce the reader to the scope and severity of the problems caused by expired domains with concrete examples. Furthermore, these examples demonstrate that many seemingly disparate security issues actually share a common underlying cause: residual trust in domains.

4.2.1 Expired Nameserver Domains

In our first example, one of the DNS nameserver domains for the Benedictine University expired—potentially leaking sensitive university emails to the domain's new owners. According to our passive DNS sources, the `ben.edu` domain owned by Benedictine University used the following nameservers, among others, in 2012:

```
ben.edu.  IN NS ns1.bobbroadband.com.
ben.edu.  IN NS ns2.bobbroadband.com.
```

In other words, the hosts under `bobbroadband.com` provided secondary NS service for the university. It is common for organizations to rely on secondary DNS services from other organizations, often in different TLDs, to provide power and geographic diversity for their DNS. Consequently, the expiration of `bobbroadband.com` did not disrupt resolution of `ben.edu` as other DNS authorities were still available. Then, on October 25, 2012, the nameservers for `bobbroadband.com` were switched to the following:

```
bobbroadband.com.  IN NS ns1.pendingrenewaldeletion.com.
bobbroadband.com.  IN NS ns2.pendingrenewaldeletion.com.
```

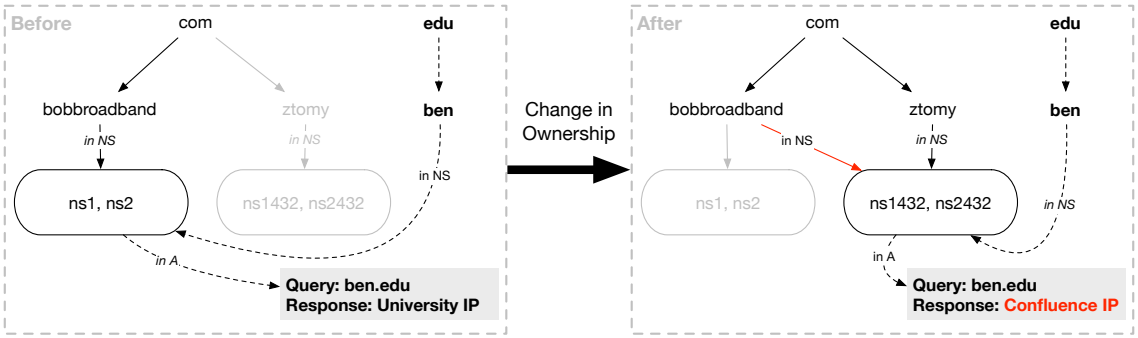


Figure 15: Residual Trust Exploitation in University DNS Servers

The zone `pendingrenewaldeletion.com` is a special zone used by the registrar to manage the final stages of the domain through to the redemption grace period. The reader should note that the redemption grace period (described in Section 2.1.3) is designed to cause an outage as a final way to notify a domain owner of an expiration. In this case, however, the redemption grace period process did not disrupt the university's DNS because other nameservers were still providing service. Ironically, the resiliency of DNS prevented the redemption grace period process from providing one last notice-through-outage to users.

After the domain expired completely, it was purchased by a search engine optimization (SEO) company that then responded to all domain queries with a wild-card answer. This directed all traffic destined for `ben.edu` (e.g., HTTP traffic, email, etc.) to an advertising site. These events are summarized in Figure 15.

This change is especially subtle because it was the domain of one of the nameservers for `ben.edu` that expired and not the university's own DNS record. Furthermore, the university still had other nameservers that would direct traffic to the school's servers, preventing the outage from occurring after every TTL for a given record. Thus, the outage intermittently manifested itself only if the nameserver handling a resolution was the one controlled by the SEO company—not one of the remaining authorities operated by the school.

Given the legal protections generally afforded to student emails, the ad company likely had no right to the traffic despite owning the domain. Clearly, there existed *residual trust* in the expired `bobbroadband.com` domain since an entire university depended upon it.

In a subsequent survey of the `edu` TLD, we identified nearly a hundred expired zones under the TLD. We offered our survey results of possible outages, similar to `ben.edu`, to the DNS community. An enterprise DNS company now provides secondary services for schools that formerly relied on expired or expiring secondary nameservers. While the problems caused in this example were many, the underlying cause was simple: *residual trust* in domains.

4.2.2 Expired Email Domains

In our second case study, we show how expired domain names could affect Regional Internet Registries (RIRs) operators. The RIRs locally administer the allocation of IP addresses [120] and maintain a database of which individuals have been allocated a specific Classless Inter-Domain Routing (CIDR) network. Stolen or hijacked RIR credentials can, therefore, lead to serious security incidents.

Account information for the RIR is protected using email as a trust anchor, and therefore, trust is effectively placed in whoever owns the domain specified by an email address. A simple check of the RIR databases yields all of the email addresses for CIDR operators, and registration checks on these domains indicated that hundreds of technical and administrative point-of-contact (PoC) listings were under expired domains.[1]

In all cases of expired contact details, we found either the `notify` or `abuse-mailbox` fields for `inetnum` and `aut-num` RIR objects contained emails under expired domains. One could simply register these domains, request a password reset, and log into the management interface to manage the allocated CIDRs. Indeed, there are several cases where this technique was abused to send spam [209].

We were in the process of notifying the various RIRs of our discovery when other researchers made public a technical report on this general problem [201]. Their work

[1]To verify the expiration of each domain, we used a domain reseller account to access the parent registry via Extensible Provisioning Protocol (EPP) [124]. This step was necessary as DNS lookups resulting in `RCODE=3` or `NXDOMAIN` merely indicate the absence of records in a zone, not the availability of the record for registration. For a discussion of EPP use, we refer the reader to [117].

focused just on RIR objects, but we believe it supports our general focus on techniques to identify and manage expired domains. We continue to work through our RIR notification process and, therefore, omit listing the affected domains.

Like the previous case study, the underlying cause of this problem is residual trust. Email is regularly used as a trust anchor for online services and email addresses fundamentally rely on domains. Consequently, possession of a domain is often sufficient to demonstrate ownership of RIR CIDR allocations.

4.2.3 Expired Browser-Related Domains

Residual trust also offers an avenue for exploiting software. For example, many browser plugins contact one or more domains on startup to load both settings and content. To quantify this problem, we inspected approximately forty thousand plugins (many with different versions) from the Mozilla store. Specifically, we examined the online credentials of the authors, sites contacted by the plugins, and the author's contact information in the XPI manifest files. We found some 159 expired domains available for immediate registration.

Anyone could register one of these expired domains used by popular web browser plugins, some with tens of thousands of installations. This creates the possibility for a new owner to push updates to the plugin or to potentially take ownership of the associated developer account. While users may have trusted the original plugin developer, this trust should not extend to the new owners of the domains used by the plugin. This problem is exacerbated by the fact that users will be unaware of such ownership changes. Given that browser plugins can modify browser settings and behavior, this leads to potential security problems that are difficult to diagnose.

Our goal here is not to simply identify another browser plugin vulnerability. Other researchers have addressed other security aspects of browser extensions [56, 57, 67, 137] by analyzing the behavior and structure of browser plugins. Indeed, our analysis of this space was aided by the tools and frameworks noted above. Rather, this case identifies yet another

instance of the unintended consequences caused by residual trust in domains. While existing work may stop potential abuse of this vector, we argue that the change in ownership of plugin domains is better dealt with by addressing the root cause: residual trust in domains.

4.2.4 Expired Open Source Software Domains

Residual trust from domain expirations also affects software repositories. Recently, the photo editing tool Gimp failed to renew its domain name, `gimp.org`. Fortunately, users noted the outage (days after the failed registration) [202] and reported the problem. This allowed the domain to be recovered during the grace period—before a malicious registrant could obtain the domain and offer corrupted versions of the software.

A more disquieting outcome was seen in the recent "Debian multimedia" episode. For a while, an unaffiliated party operated an unofficial Debian repository mirror of multimedia applications (many of which did not meet the license requirements for the official Debian distribution). The domain `debian-multimedia.org` became popular and was linked to by various blogs, HOWTO articles, and software sites. Consequently, the site was added to the Advanced Packaging Tool mirror list for many Debian users. After some discussion with the maintainers of the official Debian distribution, the `debian-multimedia.org` owner agreed to create a new domain called `deb-multimedia.org` to avoid any indication of official endorsement. The previous `debian-multimedia.org` site later expired and was registered by a party unknown to the Debian community.

In effect, the new site owner had the ability to push software updates. This capability could be used to offer updates for even non-multimedia related packages such as the kernel or the base system. While a repository key system offered users the option to protect their updates, many users may choose to ignore warnings or may not have installed a key for the old site. This risk compelled the Debian maintainers to release a warning to end users instructing them to manually remove the old repository domain [217]. The notice alerted us to the problem, which we diagnosed as yet another symptom of a larger problem: *residual*

trust in domains.

As noted above, there are protections against abuses in this dimension: software signing, local mirrors, staggered distributions in networks, rollbacks, and the like. But it is not clear if these solutions can be universally adopted by end users—many of whom simply wanted non-free multimedia software and followed well-intentioned but incomplete Internet resources. Instead of addressing the specifics of this challenging security area (the signing and verification of distributed software systems), we argue for a root-cause treatment of the problem: identifying changes in ownership of expired domains with *residual trust*.

4.2.5 Expired Spam Domains

In the previous cases studies, we examined cases where positive residual trust could be abused for malicious purposes, but we have yet to discuss the implications of domains carrying negative residual trust. Similar to benign domains, domains used for abuse often expire, and when this happens, they can be registered by new owners intending to use them for non-abusive purposes. But what happens when the new owner goes to share that newly purchased domain? Not surprisingly, the new owner may be censored by the same automatic safeguards put in place to protect online communities. Most maintainers of security lists or products will be completely unaware of ownership changes, and it may take a considerable amount of time before a domain is reclassified as non-abusive.

A public instance of this happened back in 2013 when Kirk Cameron released the film *Unstoppable*, a Christian movie targeting religious moviegoers [107]. A domain was purchased to market the film on the Internet, but this domain had previously been used to send spam—a fact presumably unknown the film's creators. Consequently, when this domain was used to market the film on Facebook, it was blocked by Facebook's automated spam detection systems. This led to heavily publicized outcries of censorship by the movie's producer and fans. Even after disclosing that the domain had been blocked by their automated spam detection systems, numerous articles decrying Facebook's censorship

practices remained without update. Such claims of censorship, even after proven false, are a risk and a liability for a social network with millions of users of differing beliefs and world views.

Ultimately, this is yet another unintended consequence of the *residual trust* placed in domains. This incident could have been prevented if there were better systems in place to evaluate the trust associated with domains. Such systems could inform potential registrants of a domain's history before purchase or update security products after domain ownership changes.

4.3 Measuring Residual Trust Abuse

In this section, we take a step back from looking at the specific cases of abuse and instead analyze the problem of residual trust abuse at scale. In particular, we analyze expired domains and malicious re-registrations from the past six years (2009–2015). We aggregate data from public blacklists, malware feeds, gTLD zone files, and other sources to measure the *scope* and *growth* of residual trust abuse. In summary:

- *Measuring Scope.* To measure scope, we identify and characterize expired domains associated with malicious behavior. In particular, we focus on expired domain names found on public blacklists or resolved by malware over the last six years. Our goal, in part, is to quantify the extent to which expired domains are exploited via malicious re-registration.

- *Measuring Growth.* For growth, we study the change in residual trust abuse over time by leveraging the temporal properties of our dataset. We measure when the domains expired and when they were used for abuse, allowing us to calculate the number of active instances of residual trust abuse.

Before diving into the results, we begin with a short discussion of the datasets used for our measurement study.

Table 9: In addition to the relative sizes of each set, this figure shows the relationships between the datasets of expired D_G, malware D_M, and public blacklist D_B domains.

Dataset Cardinalities

D_G	D_M	D_B	$D_M \cup D_B$
179,326,265	6,112,964	320,009	6,395,634

Datasets			**Dataset Intersections**		
A		B	$\%A$	$A \cap B$	$\%B$
D_G	$\cap$	D_B	0.1%	101,322	31.7%
D_G	$\cap$	D_M	0.2%	292,494	4.8%
D_M	$\cap$	D_B	0.1%	8,075	2.5%
D_G	$\cap$	$(D_M \cup D_B)$	0.2%	385,741	6.0%

4.3.1 Measurement Datasets

Restricting our observation period to 2009–2015, we focus on the domains that were *(i)* observed to expire, *(ii)* placed on a public blacklist, or *(iii)* resolved by malware. The intersection between domains that expired and that were used for abuse yields sets of domains that are likely targets of residual trust abuse—possibly resulting in a malicious re-registration. In the following sections, we define these three sets of domains and provide greater detail about their contents.

4.3.1.1 Expired domains (D_G)

We calculated the set of expired domains D_G by comparing successive gTLD zone transfers and recording removals. While the removal of a domain from a zone is a strong indicator of expiration, we further vetted such domains through the Extensible Provisioning Protocol (EPP) [124] using the domain reseller account noted in Section 4.2. Finally, we augmented D_G with data obtained from a commercial drop-catch registration service [25].

Our D_G set consists of expired domains spanning November 2008 to July 2015 and contains 179,326,265 unique domains. Most commonly, the D_G domains expired due to the registrant's failure to re-register the domain. In a few cases, the domain changed ownership due to a trademark dispute [126], suspension, or registry action stemming from a court order.

4.3.1.2 Blacklist domains (D_B)

The set D_B is an aggregation of eight public blacklists (Table 10) collected from December 2009 to July 2015. As such, it includes several different types of malicious behavior from botnets to drive-by downloads. Importantly, D_B represents a *human-curated* list of domains associated with undesirable behavior. In total, there are 320,009 unique domains in this set. We use temporal information from our sources to determine whether a domain was added to a blacklist (D_B) before or after it expired (D_G).

4.3.1.3 Malware domains (D_M)

D_M is a set of domains known to have been queried by malware. This set is compiled from three dynamic malware execution feeds: one academic and two commercial. These frameworks employ dynamic analysis to derive network and system indicators from binaries. These indicators often include URLs used for malicious purposes, e.g., command and control or advertisement fraud.

This dataset also contains temporal information for the malware execution (i.e., timestamp and DNS query), allowing us to determine whether the domain was used by malware before or after its expiration. D_M contains domains from seven years, occurring between the beginning of 2009 and July 2015, of malware execution traces from the aforementioned feeds and contains 6,112,964 unique domain names in total.

While not a guarantee of maliciousness, the domains logged by these systems adds a useful perspective to our analysis. This is especially true for those domains that appeared in a dynamic analysis trace *after* an ownership change. The reader should perceive this D_M set as an indicator, not a guarantee, of abusive behavior.

Table 10: Blacklist sources for D_B.

Blacklist	Target	Source
Abuse.ch	Malware, C&C.	[18]
Malware DL	Malware.	[26]
Blackhole DNS	Malware, Spyware.	[19]
sagadc	Malware, Fraud, SPAM.	[23]
hphosts	Malware, Fraud, Ad tracking.	[21]
SANS	Aggregate list.	[24]
itmate	Malicious Webpages.	[22]
driveby	Drive-by downloads.	[20]

4.3.1.4 Potentially abused expired domains (D_Z)

Finally, we define the set of all domains that expired and were potential targets of residual trust abuse as $D_Z = D_G \cap (D_M \cup D_B)$.[2] In the context of this study, D_Z acts as an *upper bound* on the number of expired domains witnessed between 2009 and 2015 that appeared on human-curated blacklists or that were resolved by malware. A summary describing the relationships between each of the above datasets can be seen in Table 9. In total, D_Z comprises 385,741 domains.

4.3.2 Measuring Active Residual Trust Abuse

In order to measure active instances of residual trust abuse, we focus on domains that have expired (D_G) and also appear on blacklists (D_B) or are resolved by malware (D_M). This set, D_Z, contains domains that are likely candidates for residual trust abuse through malicious re-registration of the domain. While the majority, 292,494 (75.8%), of the domains in D_Z were associated with malware resolutions, almost a third, 101,322 (31.7%), appeared on at least one hand-curated public blacklist. These numbers indicate that a substantial portion of the expired domains were manually linked with abusive behavior. This raises an interesting question. Did the expiration occur before or after abuse?

Table 11 summarizes the measurement observations behind the domain names that

[2]The Z in D_Z stands for zombie. Similarly, the G in D_G stands for graveyard. These identifiers, as well as the paper's title, are in reference to the similarities between reanimated (i.e., re-registered) domains and the depictions of zombies in popular media.

Table 11: A breakdown of how many domains expired before and after abuse for expired blacklist ($D_G \cap D_B$), malware ($D_G \cap D_M$), and all abusive (D_Z) domains—as well as the average number of days between abuse and expiration.

Expired Before Abuse

	D_Z	$D_G \cap D_M$	$D_G \cap D_B$
Num. of Domains	263,847	238,279	27,758
Avg. Days	888	911	692

Abused Before Expiration

	D_Z	$D_G \cap D_M$	$D_G \cap D_B$
Num. of Domains	123,396	54,215	73,564
Avg. Days	364	397	340

expired and also appeared in our public blacklist and malware datasets. From D_Z, we observed 123,396 domains that existed in $D_M \cup D_B$ *before* appearing in D_G. In short, these domains were used for abusive behavior before they expired. From this subset, 54,215 (43.9%) were contacted by malware and 73,564 (59.6%) appeared on public blacklists. Additionally, 4,748 (8.8%) of the domains contacted by malware also appeared on a public blacklist. Given their historical association with malicious behavior, these domains represent instances of *negative residual trust*.

Security practitioners can leverage domains with such trust for good by using them for different reconnaissance techniques like sinkholing. It is also important to note that negative residual trust can be used for malicious purposes as well. For example, an APT actor could use an expired spam-related domain to camouflage itself as a different type of threat; this would likely stymie discovery or attack attribution.

Conversely, we observed 263,847 domains that expired before appearing in $D_M \cup D_B$. More specifically, 238,279 (90.3%) domains were contacted by malware and 27,758 (10.5%) appeared on public blacklists only after expiring. Therefore, these domains represent cases of *positive residual trust* potentially being used for illicit activities. By registering expiring domains, bad actors can leverage the benefits of any positive reputation (such as brand and industry sector properties) previously held by a domain. Previously, we highlighted several concrete instances of this problem (Section 4.2). This problem is worsened by the fact that

benign domains often remain on whitelists after ownership changes due to the difficulty of discovering such events. This is highlighted by the fact that only 3,327 (1.4%) of the domains that expired before being contacted by malware ever appeared on a PBL.

To better understand the types of malware that might be abusing residual trust, we categorized some of the different types of malware observed in D_Z. Table 13 shows the top 10 malware types and families for the malware observed communicating with a simple random sample of 10,000 domains that expired and then were potentially used for abuse. Trojans are by far the most common type, with many generic types such as "malware" and "heuristic" following. The families are similarly dominated by heuristically determined labels and a few family specific labels. For example, VB.SMIS and Vobfus are generic labels for obfuscated malware written in Visual Basic. While there are instances where the MD5 is flagged as benign by the AV engines, most are malicious. As more evidence of maliciousness, 915 of the 1,559 registrars were used for registering privacy protected domain names to mask the registrant's email address and name. While there are legitimate reasons to use such a service, they are commonly employed by malicious actors to evade WHOIS attribution.

Finally, we provide a breakdown of the top-level domains (TLDs) in D_Z in Table 12. The distribution largely corresponds to the general popularity of each respective TLD. The potential exception is `edu`. We observed proportionally more `edu` domains being used for malicious purposes after expiration—possibly due to the inherent trust users place in the educational TLD.

4.3.3 Measuring Temporal Properties of Residual Trust Abuse

Next, we focus our analysis on the temporal properties of residual trust. We start by referring the reader to Figure 17, which shows the distribution of deltas between expiration and first indicator of potential abuse. On average, this delta was around a year for domains contacted by malware or appeared on blacklists. The extended length of this dormancy

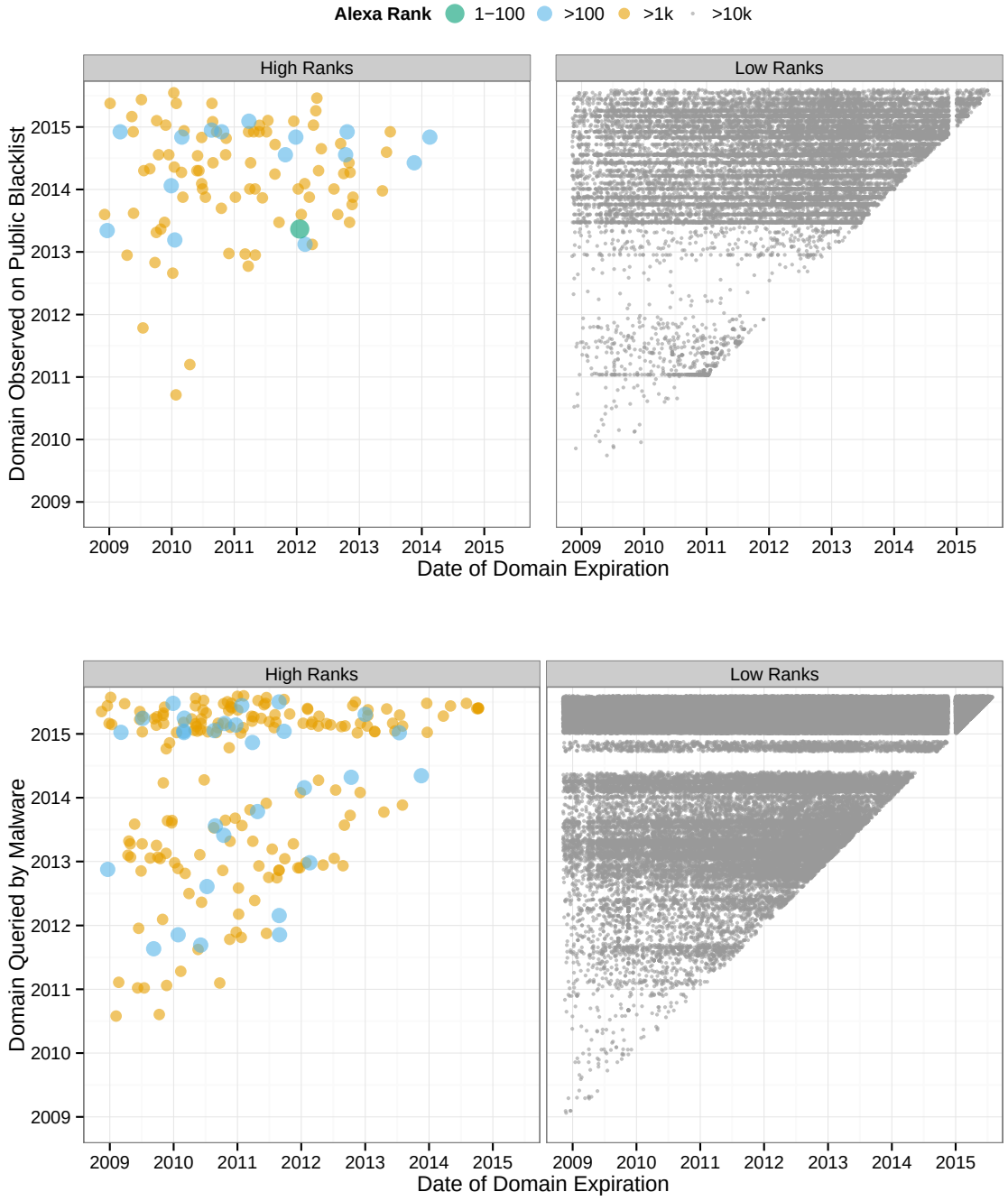


Figure 16: Expiration date of a domain versus its first blacklist appearance or contact by malware. Each point represents one of the 27,758 (27.4% of D_B) or 238,279 (81.5% of D_M) distinct domains that expired and later appeared on a public blacklist; the dot's color corresponds to the domain's Alexa rank when it was added to the whitelist. The frequency of residual trust abuse has grown by multiple orders of magnitude since we began collecting data in 2009.

period suggests that it may take a considerable amount of time before the trustworthiness of the current domain owner can be ascertained. Therefore, not only must changes in ownership be detected but such changes should be monitored until the new owner's trustworthiness can be determined.

Diving deeper into the domains that expired *before* being used for abuse, we find that the delta between the last indicator of abuse and the expiration event was roughly two years on average. The full distribution of these deltas can be seen in Figure 17 and shows two peaks, appearing approximately one year apart, for domains contacted by malware or appearing on public blacklists before expiring. The two peaks represent a small number of domains and are an artifact of shared expiration events for domains in $D_M \cap D_B$.

The long delay between last observed malware communication and expiration could be due to several factors. For example, in order to maximize the utility of malicious domains, malware authors may choose not to allow a domain to expire until the number of malicious connections to that domain drops below some threshold (i.e., the domain could still being monetized by the botmaster). Additionally, a malware author may choose to prevent a domain from expiring in order to restrict security practitioners from taking over the domain.

4.3.4 Measuring the Growth of Residual Trust Abuse

Figure 16 shows residual-trust abuse is becoming more common. The number of domains being contacted by malware after expiration grew from 6,138 between 2009 and 2012 to over 12,000 in just 2013. Similarly, the number of previously expired domains subsequently appearing on blacklists has grown from 784 between 2009 and 2012 to over 9,000 in 2014 alone. Further, more than 100 of these domains were ranked in the top 10,000 by Alexa on the day they were added to the blacklist. The horizontal striations in the figure are an artifact of malware collection and blacklisting processes. Namely, the feed operator may add many domains (possibly for the same threat) on the same day. Similarly, the vertical gap for December 2015 is the result of missing data stemming from technical issues with

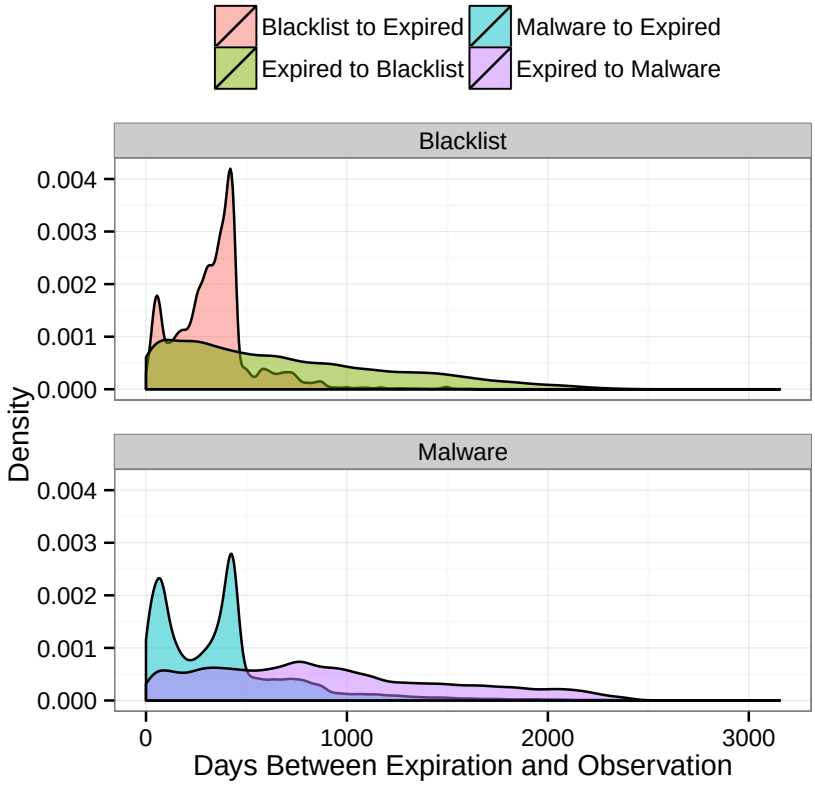


Figure 17: Distribution of the number of days between domain expiration and contact by malware or appearance on a public blacklist. This figure shows there is often a significant dormancy period before the residual trust of a domain is abused.

our collection framework.

4.4 Alembic

The previous discussion illustrated numerous studies, experiments, and anecdotes wherein expired domains with residual trust resulted in security problems or potential risks to users. While many of these cases might be remedied using well-founded research and existing technologies around phishing, DNS poisoning, and key management, it would be useful to have a system that *prevents* the problem from escalating in the first place. In this section, we leverage our previous findings to take the first steps toward such a system.

At first blush, the WHOIS protocol [28] appears to be an ideal candidate to address this question of identity. Unfortunately, WHOIS suffers from a number of limitations that

Table 12: TLD frequency for domains in D_Z. This includes all domains that were used for abuse and expired at some point. In total, we observed 13 TLDs used by these domains.

Expired to Malicious		*Malicious to Expired*	
TLD	**Count**	**TLD**	**Count**
com	214,019	com	85,409
net	27,621	net	15,954
org	9,648	info	9,287
info	5,575	org	5,869
us	2,671	biz	3,226
biz	2,185	us	2,458
ca	846	cn	989
cn	646	mobi	76
co	175	asia	56
edu	146	ca	45
mobi	80	edu	15
asia	35	co	11
de	20	de	1

make it ill-suited to deploy on a large-scale: (a) lack of verification of data, (b) expense in scaling queries across all registries and thick registrar WHOIS servers (many of whom limit queries to a handful per day); (c) lack of structure to data; and (d) lack of historical data in bulk form. We therefore explored techniques using other data more likely to be available to network operators: passive DNS logs.

The result of our efforts is *Alembic*, a general algorithm to assist in locating potential changes in domain name ownership and identifying reanimated domains. This algorithm scales without the need to mine resources such as WHOIS data and could be implemented by any network operators (or researchers) with access to DNS logs.

We measure the effectiveness of the *Alembic* algorithm using a multi-year passive DNS dataset obtained from commodity sources. As noted above, the goal of *Alembic* is to help identify potential changes in domain ownership without the expense and complexity of mining enormous volumes of WHOIS data.

Finally, we discuss two instances of residual trust abuse encountered during our evaluation. Interestingly, one of these included witnessing signs of an APT attack against sensitive networks. Finding trivially weaponizable APT domains was beyond our initial goals, but

Table 13: Top 10 malware types and families from Kaspersky/Sophos/TrendMicro for 10,000 randomly selected samples from D_M.

Type	**Count**	**Family**	**Count**
Trojan	11,979	VB.SMIS	2,449
Malware	5,436	Heuristic	2,271
Heuristic	2,271	VBCheMan-A	2,001
Worm	783	Generic	1,573
Not Malicious	486	Vobfus.M	1,014
W32	286	StartP-HV	995
Backdoor	216	Paskod-A	868
Undesirable Software	191	Not Malicious	486
Virus	86	Paskod-D	481
Packed	43	Download Agent	413

this result indicates the sensitivity of our approach and suggests other possible uses.

4.4.1 Inputs to Alembic

In order to scale and identify domains that have changed owners without expiring, we must devise an algorithm to assist in locating ownership change events, independent of WHOIS data. To achieve this, we rely on two datasets: a large passive DNS dataset to identify significant changes in infrastructure and client lookup volumes, and historic records of domain name *start of authority* to locate structural changes to the domain's zone.

There are many commercial and public passive DNS systems which collect and archive historic DNS traffic. Many organizations archive their own DNS answers, and security companies routinely maintain phone book type lists of where domains historically pointed. Readers not familiar with passive DNS may wish to consult [236]. Because we needed bulk quantities of passive DNS, we used a private collection instead of other publicly available DNS services (which often permit non-bulk API access).

Our dataset is sizable and includes historic resolutions that occurred across an entire North American ISP from January 1, 2011 to December 31, 2014. For each day, this data contains: all domain names that were resolved, the IP addresses they resolved to, and the total volume of lookups observed for a given DNS resource record (i.e., a domain name

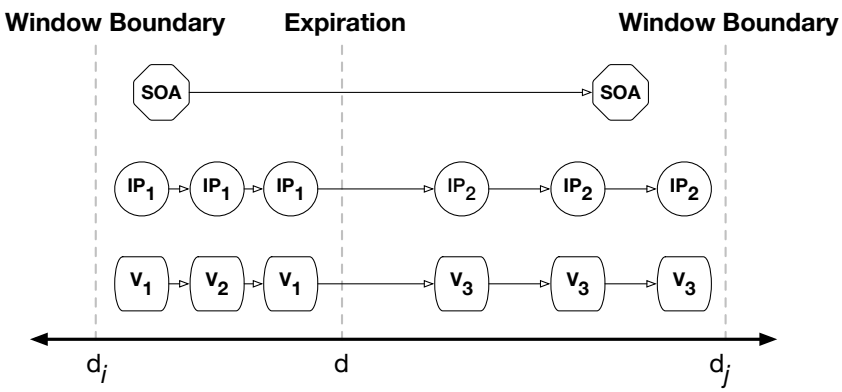


Figure 18: Using different components to identify ownership changes.

and IP address tuple). Using this data we can identify significant changes in domain name infrastructure and lookup volume—both attributes are useful for Alembic's operation.

Start of authority, or SOA, records specify authoritative information about a particular DNS zone. They contain the primary name server (the `MNAME` under RFC 1034 [164]), the email of the domain name administrator (or `RNAME` field), and other unique attributes of the zone.[3] Substantial changes in both the email and primary name server are strong indicators in ownership change for a given domain name. We therefore performed historical queries for SOA records for all the domains in D_Z.

4.4.2 Design of Alembic

We now describe how, using the aforementioned datasets, we identify domain names most likely to have undergone a change in ownership. We call our algorithm *Alembic*, after the still used by alchemists. *Alembic* lets us distill historical passive DNS evidence into a ranking of dates, and corresponding ranges, that are most likely to be associated with a change in domain ownership.

First, we discuss how we combine temporal changes in infrastructure, lookup volume, and SOA records into component scores. Then, we discuss how we generate the necessary inputs to compute these scores and how they are used to generate rankings of likely domain

[3]Those not familiar with DNS zones and DNS record types may wish to consult [232].

ownership changes.

4.4.2.1 Computing Component Scores

The *Alembic* algorithm is based upon the hypothesis that changes in ownership are highly likely to be accompanied by changes in network infrastructure, lookup volumes, and zone structure. While some users registering expired domains might be able to create the exact same zone content, host the nameservers at the same IPs, and generate the same SOA records, it is presumed this sort of subterfuge is both difficult and rare. This heuristic therefore comes down to the following conjecture: While one can perhaps buy any desired domain, one cannot so easily obtain its old IP address *and* use the same nameservers to manage the re-registered domain.

In order to identify these potential changes, the algorithm uses a temporal sliding window to measure changes in each component as observed in passive DNS resolutions over time. An overview of how the window and components fit together can be viewed in Figure 18. A summary of each individual component follows below.

Infrastructure Changes. For given a temporal window, W, we compute the Jaccard distance between hosts observed during the first and second portion of the window; this measures the dissimilarity between hosts seen during each period of time. In Algorithm 2, this measurement is computed by the INFRA-SCORE function. The computed score will range from zero to one where zero indicates the sets are exactly the same and one indicates that the two sets are completely disjoint.

Lookup Volume. Similarly, the distribution of lookup volumes for a given domain is split into two intervals for the current temporal window, W. We compute a t-test between the two distributions to measure if the null hypothesis (i.e., whether there is no relationship between them) is supported. This returns both a t-score and a p-value. The p-value ranges between zero and one with a lower p-value suggesting that the observed distributions are more likely to be consistent with the null hypothesis. Thus, a lower p-value suggests that the distributions

are more likely to be different and a higher p-value suggests that the distributions are more likely to be similar. The VOL-SCORE function in Algorithm 2 shows that the volume score is computed as one minus the p-value which results in dissimilar distributions receiving a higher score.

SOA Differences. Like the previous two cases, we compute a score based on observations about the difference between the first and second portion of the current temporal window, W. In particular, we measure changes to SOA records observed during these two intervals. Each SOA record contains two fields of interest: an authoritative nameserver, `MNAME`, and an e-mail address, `RNAME`, for the individual responsible for the zone. We measure changes to each of these fields independently in order to finely measure changes in SOA records. Thus, we compute the Jaccard distance between the set of `MNAME`s observed in each portion of W, and separately, we compute the Jaccard distance between the set of `RNAME`s observed in each portion of W. The SOA-SCORE function, in Algorithm 2, shows how we compute the overall score for changes in SOA records, and like the previous component scores, higher values indicate there were more changes between the first and second portion of the temporal window.

4.4.2.2 Alembic Algorithm

The *Alembic* algorithm uses the component scores to generate rankings of likely domain ownership changes. Algorithm 3 presents a pseudo-code implementation of the *Alembic* algorithm.

The first step in the algorithm is to choose a window W. This window defines the number of days worth of passive DNS data, around some date d, required for the algorithm to compute a change in ownership score. For example, if $W = 14$, then seven days worth of records before and after d are necessary for the algorithm to run; if insufficient records are available, the algorithm simply returns zero. In Algorithm 3, this process results in h_i and h_j, which are sets of hosts seen in A records $\frac{W}{2}$ days before and after d. These sets are used

Algorithm 2: Computing Component Scores()

function INFRA-SCORE(h_i, h_j)**:**
  return *1* - JACCARD-INDEX*(h_i, h_j)*
end

function VOL-SCORE(v_i, v_j)**:**
  $t\_val, p\_val \leftarrow$ TTEST(v_i, v_j)
  return $1 - p\_val$
end

function SOA-SCORE(s_i, s_j)**:**
  $m_i, r_i \leftarrow s_i$
  $m_j, r_j \leftarrow s_j$
  $M \leftarrow \frac{1}{2}(1 -$ JACCARD$(m_i, m_j))$
  $R \leftarrow \frac{1}{2}(1 -$ JACCARD$(r_i, r_j))$
  return $M + R$
end

as the input to INFRA-SCORE to compute the infrastructure component score.

Since not all domains will have W contiguous days worth of records around d, the algorithm tries to pick the $\frac{W}{2}$ closest days before and after d. This may result in date ranges of varying size for each half of W. Therefore, we compute the date range for a window, W, by finding the minimum date, d_i, associated with the records in h_i and the maximum date, d_j, associated with the records in h_j.

We use the date ranges $[d_i, d]$ and $(d, d_j]$ to compute the lookup volume distributions for each portion of W around d. If we do not have lookup volumes associated with a date in one of these ranges, we assign it a lookup volume of zero; this imbues information about how frequently the given domain is resolved. The lookup volume distributions for each date range, v_i and v_j, are given as inputs to the VOL-SCORE to compute the lookup volume component score.

Next, the SOA records observed between the date ranges $[d_i, d]$ and $(d, d_j]$ are placed into two sets, s_i and s_j, and these sets are given as parameters to SOA-SCORE to compute the SOA component score.

Algorithm 3: Alembic Algorithm

function ALEMBIC(d, h, v, s) **:**
  $W \leftarrow$ window size
  if $|h| \geq W$ **then**
    $h_i \leftarrow \frac{W}{2}$ records before date d in h
    $h_j \leftarrow \frac{W}{2}$ records after date d in h
    $score_h \leftarrow$ INFRA-SCORE(h_i, h_j)

    $d_i \leftarrow$ minimum date for record in h_i
    $d_j \leftarrow$ maximum date for record in h_j

    $v_i \leftarrow$ lookup distribution between $[d_i, d]$ in v
    $v_j \leftarrow$ lookup distribution between $(d, d_j]$ in v
    $score_v \leftarrow$ VOL-SCORE(v_i, v_j)

    $s_i \leftarrow$ SOA records seen between $[d_i, d]$ in s
    $s_j \leftarrow$ SOA records seen between $(d, d_j]$ in s
    $score_s \leftarrow$ SOA-SCORE(s_i, s_j)

    return $score_h + score_v + score_s$
  else
    return *0*
  end
end

Finally, the change of ownership score is computed as the sum of each component score, which results in a value that ranges between zero and three. This score should be computed for each date that a passive DNS resolution was seen for a domain; these scores can then be sorted from highest to lowest to provide a ranking of dates, and corresponding ranges, which are most likely associated with changes in domain ownership.

The resulting list can be used to provide additional information about domains based on their residual trust. For example, whitelists can be pruned so that benign sites undergoing an ownership change can be quickly remapped to another appropriate category (e.g, "unknown" or "untrusted") depending on the context. Knowledge of ownership changes can be leveraged to improve existing reputation and detection systems.

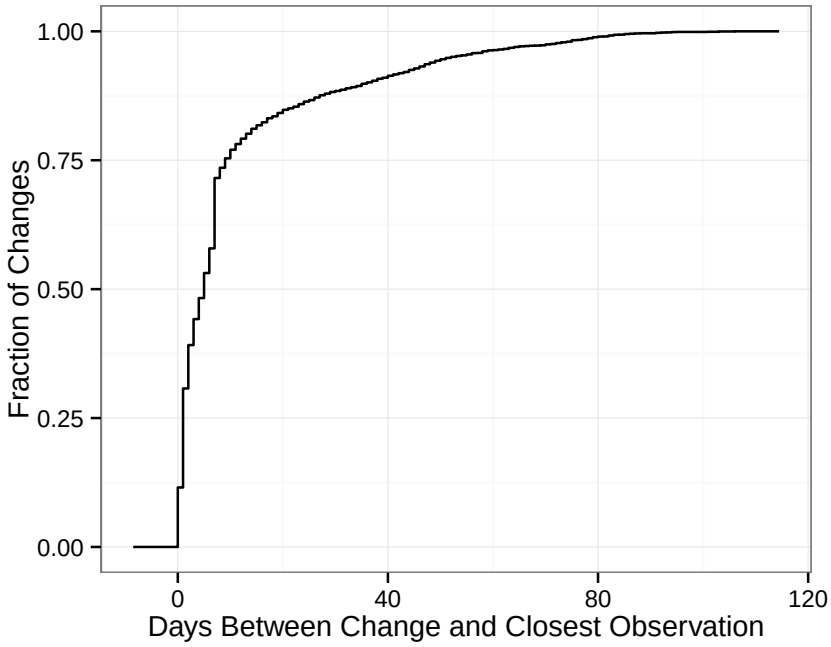


Figure 19: CDF showing the distance (in days) between an ownership change and the closest observation in our passive DNS dataset. For 75% of the ownership changes, there is an observation in the passive DNS dataset that is less than 20 days away.

4.4.3 Efficacy of Alembic

Using the Alembic algorithm and our passive DNS dataset, we compute the ownership scores for a sample of *active* domains in D_Z. In our analysis, we define a domain as active if it was resolved at least W, with $W = 14$, times over any 120 day period in our dataset. This requirement filters domains for which the lack of observations would yield unreliable results. Similarly, we restrict our analysis to domains for which we were able to acquire ground truth about ownership changes. In total, we calculated 764,681 ownership scores for 11,564 domain names.

We compared the scores against known ownership changes gathered from archives of historically collected WHOIS data [29]: 17,838 changes in total. Figure 19 shows the distance between actual date of change and the closest observation date in our dataset. In short, 80% of the confirmed changes fall within 13 days of an observation in our dataset. This result is important as the effectiveness of Alembic depends on the frequency of DNS

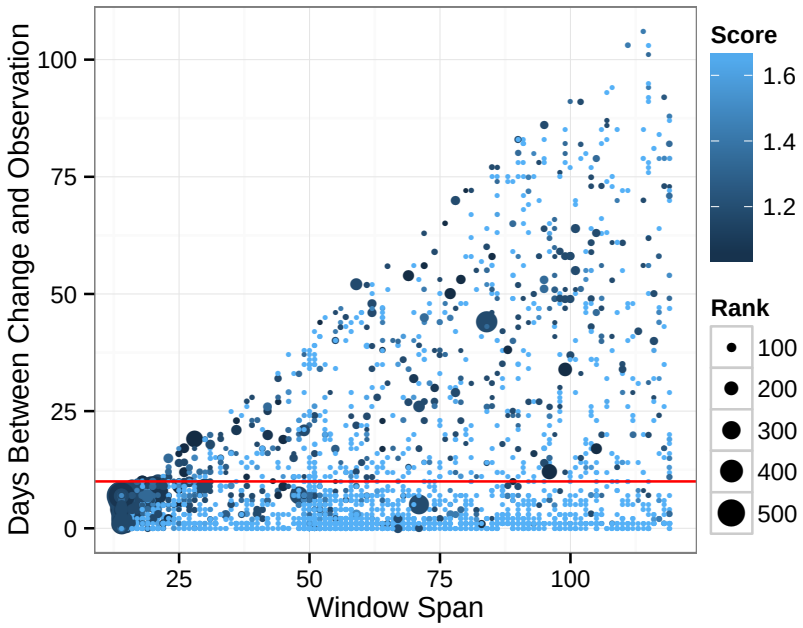


Figure 20: Window timespan required for W observation days versus the distance between date of change and closest observation. This figure shows the best Alembic can perform given the sparse nature of the DNS resolutions for the domains in D_Z.

resolutions for a domain. Specifically, Alembic requires at least $W/2$ observation days before and after the candidate date. In other words, the span of the observation window depends on the resolution frequency of the domain. At a minimum, the window may span W consecutive days, i.e., the domain saw a DNS resolution on all W days. In the worst case, the domain may only be resolved once over our dataset's collection period. As mentioned above, we cap the date range necessary to collect W days of resolution behavior at 120 days. We show the date range of the observation window with respect to the number of days away from an exact match in Figure 20. In total, we find 4,543 (25.5%) of all changes fall within an Alembic observation window. Encouragingly, the bulk of these ownership change events occurred within ten days of an observation (red line in Figure 20)—even for larger observation ranges.

We believe our algorithm is a necessary step towards fostering additional research into domain ownership changes. Furthermore, our results show that Alembic, which works

without relying on archiving and parsing WHOIS records, identifies potential changes in ownership. We plan to improve and refine Alembic to account for multiple ownership changes and sparsity in the input DNS data. For the latter, we propose investigation into the relationship between the frequency of resolutions for a domain and the span of the observation window required to detect ownership changes. Finally, we plan to explore other detection signals to use as component scores.

4.4.4 Additional Discoveries Using Alembic

We used Alembic to help identify abuses of both positive and negative residual trust. Here we discuss examples that fall into each of these categories. For the former, we highlight previously benign domains which were later used for command and control (C&C), leveraging the domain's historic reputation to exploit whitelisting. For negative residual trust, we highlight a potential attack vector whereby a leftover domain from a state-sponsored threat could be used to trivially gain access to sensitive networks where an infection has already occurred.

4.4.4.1 Abuse of Positive Residual Trust

Here we study cases where Alembic helped identify cases of positive residual trust abuse. We present a brief look at two of the 263,847 domain names that were located by Alembic and subsequently became malicious only after expiring.

First we look at `doctorcompany.net`. After expiration, malware began using this domain for command and control (C&C). Anti-virus analysis from VirusTotal suggests this particular malware was variant of Win32/Polif [15] (a.k.a. Symmi). This particular threat is capable of numerous malicious activities including downloading and executing arbitrary files, logging keystrokes and other sensitive data, and exfiltrating any stolen information.

Using available historic WHOIS data, we estimate that `doctorcompany.net` changed owners once between 2008 and 2014. As shown in Table 14, the new owner chose to use an identity protection service when registering the domain, a common tactic used to by

Table 14: Ownership changes to `doctorcompany.net`

Date	Reg. Name	Reg. Email
10/15/08	Marcos Paulo dos Santos Fortunato	marcos.fortunato@contato.net
02/07/13	Identity Protection Service	doctorcompany.net @identity-protect.org

both legitimate and malicious users to exclude personal information from WHOIS records. Throughout the second lifetime of the domain and until its expiration—listed in the WHOIS record as February 7, 2014—the domain used the same nameservers, suggesting the owner remained the same during that year. We confirmed the domain became available for registration again on April 29, 2014—81 days after the listed expiry date and long enough to have passed through the entire expiration process described in Section 2.1.3. About a month later on May 25, 2014, we saw malicious binaries attempting to query this domain. Since this domain had approximately six years of history without abuse, subsequent use by malware benefited from the domain's positive residual trust.

Similarly, `clicky.info` was also used for malware command and control (C&C) only after domain expiration. AV analysis suggests this particular malware sample is a variant of Win32/Nivdort [27], a trojan that steals key-presses, browsing history, credit card information and user-names and passwords. Using historic WHOIS, the domain's ownership appears to have changed eight times over the course of eight years. A summary appears in Table 15. Using historic WHOIS, we were able to confirm this domain was in pending delete status on February 11, 2014, and we subsequently confirmed its expiration on February 13, 2014 using the techniques mentioned in Section 4.3.1. As seen in Table 15, it was subsequently seen re-registered on March 9, 2014. The first observed communication by malware to this domain occurred on March 15, 2014—less than a week after being re-registered. Consequently, malware using this domain is able to leverage almost eight years of positive residual trust.

The WHOIS data for `clicky.info` shown in Table 15 also highlights that that ownership changes are not always preceded by an expiration (domain registrations typically

Table 15: Ownership changes to `clicky.info`

Date	Reg. Name	Reg. Email
03/16/06	Kim Fisher	jadothebest@hotmail.com
03/23/07	Derek Giordano	Derek@generalrate.com
01/01/09	Anders Oie	anders_oie@hotmail.com
04/05/10	Rubalier	cvx.conts@gmail.com
10/20/10	barry harding	bharding777@gmail.com
11/30/12	WANG SONGXU	sdwildcat@163.com
11/26/13	del del	del@del.del
03/09/14	Jeffrey Aikman	Roldvale@aol.com

last at least one year). This further motivates the need for an algorithm like Alembic that helps locate ownership changes and illustrates the need for better awareness around the abuse of residual trust in domains.

4.4.4.2 Abuse of Negative Residual Trust

Next, we highlight a potential attack vector that leverages expired APT domains. On June 9, 2014 the security company CrowdStrike publicly released a report [14] detailing the cyber espionage activity of PLA Unit 61486. Also known as PUTTER PANDA, Unit 61486 is a branch of the Chinese SIGINT community.[4] Their mission, according to CrowdStrike, is to steal the trade secrets of corporations in the satellite, aerospace, and communication industries.

CrowdStrike's report identifies Chen Ping, as the primary persona responsible for obtaining domains for Unit 61486's C&C infrastructure. This moniker was derived from the registrant email stored in the WHOIS records, `cpyy.chen@gmail.com`. We leveraged this knowledge to identify `usreports.net`, an expired domain in our dataset that was previously registered using Chen Ping's email. We reanimated the domain, pointed it to a sinkhole, and found that despite being expired for years (and Unit 61486's activities being publicized in high-profile white-papers) our sinkhole began receiving connection attempts, every three seconds, from a national government research lab in Taiwan.

[4] Unit 61486 is distinct from Unit 61398 described in Mandiant's APT1 report [10].

It follows that any malicious party with knowledge of the C&C protocol can capitalize on expired C&C domains to gain entry into already compromised networks—all for the low price of domain registration. This raises an important question: Should domains be available for re-registration after they were previously used for malicious purposes? We discuss this issue more in the following section.

4.5 Discussion of Potential Remedies

Throughout this study, we have highlighted malicious re-registration and residual trust as the root cause of many seemingly disparate security problems. In Section 4.2, we outlined several attacks and security lapses made possible by the abuse of this residual trust. Current solutions only address the symptoms of the underlying problem, not the cause, resulting in a plethora of techniques that only address narrow avenues of abuse. Instead, these problems would be better solved by addressing the underlying abuse vector.

In this section, we discuss potential remedies, both non-technical and technical, for residual trust abuse. Unfortunately, there is no single solution that can completely solve the problem; instead, a comprehensive remedy necessitates discussion and cooperation between all affected stakeholders. Our analysis of remedies is intended to outline the challenging nature of the problem with the hope it will foster further investigation by the security community.

4.5.1 Non-Technical Remedies

While any domain may carry residual trust, the severity of potential abuse is much greater for certain types of domains, e.g., those previously used by financial institutions or critical infrastructure. In short, domains that affect large numbers of users and systems, if abused, would benefit more from greater protections than other less important domains.

One potential remedy is to restrict critical industries to specially regulated zones. The idea is to limit who can register expired domains from one of these protected zones. Indeed, we already see this type of behavior with zones like `gov` and `edu`. Unfortunately, there

are several unresolved questions and challenges with this solution. First, what criteria must be met for a domain to be considered critical? Second, how do we identify the existing critical domains? Third, assuming such domains could be identified, how do we migrate each domain from its existing zone? Finally, who is responsible for creating and managing the critical zones? These questions are made even more complicated by the global reach of the Internet; many diverse organizations (with different goals and motivations) would need to reach a consensus before any global solution could be adopted.

Rather than rely on custom zones, another potential option is to have the registrars or registries enforce special registration policies for critical domains. This solution is attractive as it could provide protection to critical domains under all zones and not simply those under a special top-level domain. However, this requires identification and reporting of all critical domains to either the registrars or registries and, for many organizations, this could be a challenging task. It also does not solve the problem of which domains qualify for protected registrations. This solution may be further complicated by the fact that any solution involving the registrars or registries also presumes that they would be willing participants. Given their financial interest in selling domains, there is a strong possibility that they would be reticent to employ any policies that make domain registration more cumbersome.

The previous two solutions focus on identifying critical domains; however, such solutions do not address the case where a non-critical domain is used as a trust anchor. For example, in Section 4.2.2 we saw how email addresses for expired domains were used for account management, thereby opening up the possibility for an attacker to hijack the account using malicious re-registration. For these domains, non-technical remedies need to be augmented with technical ones; we will discuss a couple such options in detail below.

4.5.2 Technical Remedies

When non-technical remedies fail, a technical solution is needed to mitigate problems. There are innumerable services that rely on third party domains, either for infrastructure or from

users, and it is unlikely that many of these domains would fit some strict definition of a critical domain. As such, the non-technical policies proposed above would not be sufficient.

Instead, these systems should employ some process, such as Alembic (Section 4.4), for identifying potential ownership changes. Such changes should be used to expire or revise the inherent residual trust of the associated domains. For instance, systems that rely on e-mail should re-evaluate access policies when e-mails expire or change ownership. A firewall rule that whitelists a domain should be revised to reclassify domains in order to avoid missing new attacks. A security information and event management (SIEM) device that classifies a domain as "low risk/spam/click-fraud/SEO" may revise the scoring of domains that have changed ownership. Given the active role of expired domains in APT attacks, this recommendation applies equally to forensic analysts and those investigating post-compromise events.

For smaller numbers of domains, it may be possible to use WHOIS to identify when the residual trust of domains should be re-evaluated, but this will not scale due to the complexities of bulk WHOIS collection. Furthermore, the lack of consistent formatting, use of privacy protection services, and inconsistent verification of WHOIS data may cause inferences relying on it to be unreliable. A system like Alembic could be used to address some of those concerns. In particular, it could be used to help identify ownership changes when scaling WHOIS becomes untenable, and since it relies on underlying network properties, it may find ownership changes that would be missed in WHOIS due to unreliable or forged data.

Dealing with residual trust is a challenging problem, but ignoring it exposes users and systems to a host of security issues. A comprehensive solution for this problem will require additional research and discussion by the security community.

4.6 Summary

Domains can change ownership for many reasons (e.g., expirations, auction, transfers) and the remaining *residual trust* is abused by clever attackers hoping to evade whitelists, hijack accounts, exploit software systems, or even buy access to existing infections. In short, we find that residual trust abuse is the root cause of many security issues on the Internet. At its core, there are potential policy and technical remedies. Policy remedies could identify potential avenues for exploiting residual domain trust and prevent or police re-registrations as appropriate. When that fails, technical remedies should actively try to identify ownership changes; we propose one such algorithm, Alembic.

Using a dataset of 179,326,265 expired domains spanning from December 2008 to July 2015, we quantify and characterize residual trust abuse and malicious re-registration. We found that 385,741 expired domains were contacted by malware or appeared on a public blacklist. This intersection contained almost a third, 101,322 (31.7%), of public blacklists domains in our dataset, and more troubling, a little over quarter, 27,758 (27.4%), of these domains expired before being blacklisted. In addition, only 3,327 (1.4%) domains contacted by malware after expiration ever appeared on a public blacklist. These findings demonstrate that the residual trust of expiring domains is being actively exploited. To make matters worse, we observe that the number of domains showing up on blacklists after expiration has grown from 784 between 2009 and 2012 to over 9,000 domains in 2014 alone; this shows that residual trust abuse is a growing phenomenon.

In order to help the research community flag potentially dangerous reanimated domain names, we developed a lightweight algorithm to rank potential domain ownership changes using only features that can be passively collected from DNS. We used this algorithm to identify several cases of residual trust abuse; specifically, we identified instances where re-registered domain names were used as infrastructure to facilitate attacks and one instance where an expired APT-related domain name could have been re-registered to gain access to an overseas government research lab.

CHAPTER V

EMPIRICAL ANALYSIS OF TRADITIONAL MALWARE

5.1 Motivation

Malware analysis is at the forefront of the fight against Internet threats. Over the last decade, numerous systems have been proposed to statically and dynamically analyze malicious software and produce detailed behavioral reports [169, 244]. The vast amounts of data collected by such systems can be used to provide important reputation information about both IP and domain name system (DNS) infrastructure, which play an important role in the state-of-the-art detection engines used by the security industry.

Despite the fact that an increasing number of companies and researchers now have access to large malware databases—often containing millions of samples—little is known about how the infrastructure and methods used by Internet miscreants has evolved over time. Previous studies [105, 171, 173, 216, 248, 251] often used small datasets and performed very specific analysis—focusing on topics like the role of cloud providers, the infrastructure behind drive-by downloads, or the domains used by few malware families.

To shed light on this important problem, we performed a five year, longitudinal study of dynamic analysis traces collected from multiple (i.e., two commercial and one academic) malware feeds. These feeds contain network information extracted from the execution of more than 26.8 million unique malware samples. We complement this dataset with over five billion DNS queries collected from a large North American internet service provider (ISP). The combination of these two sources provides a unique view into the network infrastructure that malware samples have contacted over the past five years.

There are three main differentiators between this work and all the previous work that we build upon. First, we analyze several orders of magnitude more data than any prior research

efforts, and we do so over a much longer observation period—almost five full years. This affords us unique insights into how tens of millions of malware samples have evolved over time. Next, we link network level communications (e.g., domains and IPs) with system oriented information (e.g., malware families, PUP). Most existing work does not attempt to perform both these types of analysis in concert—let alone at this scale. Finally, we provide temporal analysis of malware communication over time. This gives us interesting insight into the relationship between the first observable network communication and discovery of malware by the community.

It is only fair to acknowledge that our work was possible because of the many prior efforts in the fields of malware and network analysis—the most notable of which we cite. The fact that our findings confirm the results of many past studies lends further weight to their results and serves to make them more generalizable. We believe the community needs a combination of both large-scale longitudinal studies and more focused small-scale studies. The former better captures global phenomena and general trends while the latter enables more detailed investigations by allowing for manual analysis and deeper inspection of traffic.

5.1.1 Contributions

Our study resulted in four different contributions that we will discuss in greater detail below.

First, while dynamic analysis traces can be used as ground truth and forensic evidence of an infection, they should be very carefully curated. Conducting our long term analysis required us to devise a comprehensive filtering process to remove benign domains from our datasets. This process emphasizes the challenges of reducing the inevitable noise present in any large dataset. We provide (Section 5.3) a detailed and extensive set of rules that network defenders should follow when they wish to remove potentially benign domain names from their dynamic analysis traces.

Next, we observe that PUPs are not only on the rise (Section 5.4) but also that they surprisingly utilize a very stable network IP infrastructure. Our analysis shows that PUP

families host their infrastructure on popular cloud hosting providers and CDNs for up to several years. This may indicate that popular hosting providers do not have the same abuse policies towards banning PUPs that they use to fight malware. This analysis required us to modify an existing clustering tool [203] and perform the largest malware classification effort to date.

Third, dynamic malware analysis traces are far from the ideal source of information for building early warning systems or detecting new emerging threats. In our analysis, we see that domain names used in malware communications are active weeks, sometimes even months, before malware gets discovered and analyzed by the security community (Section 5.5.2.2). This observation has a direct implication on malware domain name blacklists (Section 5.5.1.1). While they are certainly useful for detecting current and past malware families, they are not necessarily an efficient method of combating future malware threats. In fact, our long term study shows (Figure 26) that malicious domains were added to major blacklists several days after the malware appeared in one of our feeds and months after the potentially malicious communication was seen in passive DNS.

Lastly, we study the evolution of the IP infrastructure resolved by malware and PUP domains over time, and we identify three interesting categories of "hot spots" in the IP space. These categories correspond to (1) IPs associated with large families that use the same network for extended periods of time, suggesting significant deficiencies in current network and system level defenses; (2) IPs associated with sinkhole operations run by security organizations; and (3) IPs associated with hosting providers that are more willing to tolerate malicious infrastructures, resulting in frequent use by several families. We also analyze the roles of dynamic DNS (DDNS) and content delivery network (CDN) services, as they are both frequently used by malware, and show that approximately 32% of all malware samples in our dataset queried at least one dynamic DNS domain. Finally, we measure the prevalence of domains created by domain generation algorithms (DGAs) in network communication from malware samples, and we find that at least 44% of the domains from

Table 16: Summary of datasets used. All datasets correspond to January 2011–August 2015.

Dataset	Data	Count
Malware Executions	Samples with DNS	26.8 M
	FQDNs	11.5 M
	e2LDs	6.8 M
	IPs	1.4 M
VirusTotal	Reports	23.9 M
Passive DNS	Resource Records	5.2 B
	FQDNs	4.6 B
	e2LDs	2.9 M
	IPs	178.7 M
Public Blacklists	Distinct Blacklists	8
	e2LDs	320 K
Alexa	e2LDs	8 M
Expired Domains	e2LDs	179 M
DGArchive [31]	DGA FQDNs	50 M

dynamic malware traces are generated by 42 DGA families

5.2 Datasets

Table 16 summarizes the datasets used in this work. All data corresponds to the time period from January 1st 2011 to August 31st 2015 unless otherwise noted. We use three malware executions datasets to obtain the domains resolved by malware and the IP addresses they resolved to; a passive DNS dataset to map domains to IP addresses and obtain an estimation of their query volume; VirusTotal (VT) reports to obtain additional metadata for the executed malware; public blacklists to identify dates when malicious domains were blocked; the historical Alexa top 1M for whitelisting benign domains; domain expiration dates to mark end of ownership events; and the DGArchive [31] to identify DGA domains. Each of these datasets is described in more detail below.

In this paper, we focus on effective second level domains (e2LDs) rather than fully qualified domains names (FQDNs) because e2LDs better capture domain ownership. For example, the FQDN `www.google.com` has e2LD `google.com`, while `www.amazon.co.uk`

Table 17: Summary of the public blacklists used in this study.

Blacklist	Target	Source
Abuse.ch	Malware, C&C.	[18]
Malware DL	Malware.	[26]
Blackhole DNS	Malware, Spyware.	[19]
sagadc	Malware, Fraud, SPAM.	[23]
hphosts	Malware, Fraud, Ad tracking.	[21]
SANS	Aggregate list.	[24]
itmate	Malicious Webpages.	[22]
driveby	Drive-by downloads.	[20]

has e2LD `amazon.co.uk`, since the second level domain `co.uk` does not correspond to the domain owner. Thus, unless otherwise noted, when we talk about domains we refer to e2LDs and only use FQDNs for better differentiation when needed.

Malware Executions. We collected all the domain names resolved by malware samples from three different datasets—each containing the MD5 of the malware, date of execution in the sandbox, domain names resolved during the execution, and IP addresses that domains resolved to. Each malware sample ran for no more than five minutes in each of the different datasets.

We briefly describe the three datasets but will only refer to their union, after removing duplicate samples, throughout the rest of the paper .

- $UNIVERSITY$. This dataset comes from a university-operated malware execution environment. Collected from January 2011 to August 2015.
- $VENDOR$. This dataset comes from the malware execution environment of a large security vendor that tracks spam and e-mail abuse. Collected from September 2014 to August 2015.
- $ANUBIS$. This dataset comes from the Anubis Web service [146], where users can upload suspicious samples for dynamic analysis. Anubis has operated since 2007, but we focus on executions between January 2011 and June 2014.

In total, we collected the network behavior of 26.8M unique malware samples. It is important to note that this number excludes samples without any valid or successful DNS resolutions.

VirusTotal Reports (VT). VirusTotal [37] is an online service that analyzes files and URLs submitted by users. Submitted executables are scanned with multiple AV engines. VT offers an API to query meta-data on malware samples using a sample's hash, and we queried VT using the 26.8M hashes. For each sample, we collected the time it was first observed by VT, AV analysis date, and AV detection labels. Of the 26.8M samples, 89% were known to VT at the time of our submission (i.e., during the period 2015-16).

Passive DNS ($pDNS$). Due to agreements with the provider of this data, we cannot publicly disclose the exact source, but we can state that this dataset contains passive DNS data collected from a large ISP in the United States. It contains the domain names resolved by clients of the ISP and the IP addresses those domains resolved to. This data was collected above the recursive DNS server, and therefore, it does not contain information about the clients making requests—rather it aggregates resolutions from all clients. In particular, the dataset contains resource records (i.e., timestamp, queried domain name, and associated RDATA [164, 165]), as well as domain lookup volumes aggregated on a daily basis. It comprises 2.9M e2LDs resolving to 178.7M IP addresses.

Public Blacklists (PBL). This dataset contains 320K malicious e2LD entries extracted and aggregated from the eight public domain blacklists, detailed in Table 17, which we regularly collected and updated for the entire duration of the project. Due to this aggregation, the dataset includes multiple types of abusive domains such as drive-by downloads, phishing, and botnet C&C. These domains are curated by members of the security community and, thus, represent cases of human verified abuse. For each domain, the data also provides the exact date when the domain was included in the blacklist.

Alexa. This dataset contains rankings of the Alexa top million domains collected daily [40]. It contains approximately 8M unique e2LDs across our entire analysis period.

Expired Domains. This dataset includes the expiration dates of 179M (benign and malicious) e2LDs for the past seven years. These expirations were verified by recording removals from successive gTLD zone transfers and, since the removal alone does not always indicate an expiration, were further vetted using the Extensible Provisioning Protocol (EPP) with a domain reseller account. This methodology is modeled off of previous work that studied potential pitfalls resulting from domain ownership changes [153].

DGArchive. Plohmann et al. [182] recently reverse-engineered malware families that use a DGA. The results of their work is collected in the DGArchive [31], a database of 50M domains that can be generated by the DGAs of 66 malware families. We use the DGArchive to identify DGA domains among the domains resolved in the malware executions.

Limitations and Potential Biases. Despite our best efforts to collect the most comprehensive set of data sources to perform our study, there are still some limitations and potential biases worth mentioning. For example, our study cannot cover samples that have failed to run or that used evasion techniques to avoid revealing their network behavior in the analysis sandbox. To ameliorate this issue, we combine three different malware feeds each using their own sandbox environment. Our datasets also have some geographical bias towards the United States, since the passive DNS data was collected from a large US ISP. However, we believe some form of bias is unavoidable in this type of study. Compared to the state of the art in DNS and malware analysis, our datasets still provide the broadest and deepest view on malware network behavior to date by far.

5.3 Domain Filtering

We start our analysis by processing the DNS requests performed by malicious files run in dynamic analysis sandboxes. For this, we first remove 255,747 samples that were not flagged as malicious by any AV vendor (according to the results collected in our VirusTotal dataset). What was left was a set of likely malicious or unwanted files.

Since malware does not interact with exclusively malicious infrastructure, not all domains

queried by malware samples can be considered malicious. In fact, as it is often the case with large datasets, the initial set of DNS requests was very noisy and needed to be carefully pre-processed to remove all spurious and unwanted entries. In our study, we want to focus on domains that are associated with actual malicious communication. However, despite the fact that all domains are requested by malicious files, the vast majority of the requests in our dataset did not fall in this category. While this may seem surprising at first, it is the consequence of several factors—such as the presence of non-existent domains generated by domain generation algorithms (DGAs), connectivity tests to benign domains, sinkholes, and spam-related activity.

To remove this noise, we proceeded with an initial filtering phase divided into four separate steps with the goal of eliminating invalid, benign, or sinkholed domains as well as reverse delegation queries.

Invalid Domains. Since not all DNS requests result in a valid resolution, we first filter out DNS queries that request non-existent domains (i.e., that do not return a valid IP address). This step is particularly important to reduce the impact of domain generation algorithms (DGAs), where malware tries to resolve many possible domains until it finds one that has been registered by the botmaster. We study the resolutions for non-existent domains, which may be subsequently registered and used for abuse, and DGA behavior in Section 5.7.

Overall, this first filtering step successfully reduced the number of unique effective second level domains from 6,850,793 to 1,316,331 , and the number of fully qualified domain names from 11,532,653 to 3,767,234 .

Benign Domains. The hardest part of domain filtering consists of identifying and removing the queries performed towards benign domains. Their presence is due to many factors that include malware using legitimate services (e.g., Dropbox), testing if the infected machine has a working Internet connection, downloading components from compromised websites, delivering spam messages to victim mail servers, and even querying an existing benign domain as a result of collisions in a poorly designed DGA algorithm.

The variety of potential causes makes it very difficult to automatically filter out all benign DNS requests. Our approach relies on three separate steps. In the first, we use the $ALEXA$ dataset to remove domains that appeared in the Alexa top ten thousand most popular domains for at least a year, with the exception of dynamic DNS domains—which are often abused for malicious purposes. While the $ALEXA$ dataset provides a good starting point, it fails to capture some obviously popular domains. Therefore, in the second step we manually sifted through the most popular domains remaining after the Alexa filtering, and we identified and removed from our dataset other popular sites such as content distribution networks. This step reduced the set of effective second level domains from 1,316,331 to 1,291,313 and fully qualified domain names from 3,767,234 to 3,295,860.

Finally, we noticed that the remaining dataset was largely dominated by spam bots, which query hundreds or even thousands of benign domains with the goal of locating the SMTP servers of their targets. A comprehensive study of spam behavior is outside the scope of this study. Therefore, we used an aggressive filter that removed any samples performing MX lookups and, as some malware may receive a pre-generated list of MX records, samples that queried for domains containing mail-related keywords (e.g., mail, smtp, imap). While excluding entire samples matching this filter may seem aggressive, we observed that only 405,742 (1.5%) distinct samples contained at least one MX or mail related domain. The presence of these domains suggests a different type of behavior from the rest of the samples in our dataset, and therefore, we chose to discard them to avoid missing less popular, benign domains they may have queried.

In total, their removal reduced the set of effective second level domains from 1,291,313 to 329,348 and fully qualified domain names from 3,295,860 to 2,154,609.

Reverse Delegation Zones. DNS Pointer Records (PTR) often reflect activity from system processes (e.g., gethostbyname()) trying to resolve IP addresses in a remote network. This can occur when a program directly connects to an IP address without performing a DNS resolution of a service's domain name. For example, Windows logging makes note of a

network socket connection but avoids listing the IP address—generating a DNS PTR record instead. This behavior, associated with Windows logging of RFC 1918 [170] host names, can be observed at the root levels of DNS [63]. Thus, dynamic execution of a malware may generate reverse delegation domain names that point to remote residential IP space. While the IP could be malicious, the reverse delegation domain name and its effective second level domain cannot be considered malicious as they are typically owned by the ISP (e.g., Verizon) or the hosting provider (e.g., Rackspace).

While it may seem reasonable to remove all e2LD domains seen in PTR records, this would result in too coarse of a filter because the owner of the netblock has the power to assign any domain as the reverse DNS pointer. Thus, some PTR domains will contain the actual domain name used to resolve an IP address instead of a domain, created by the ISP or hosting provider, to describe the underlying infrastructure.

In our final step, we remove benign PTR domain names from our malware domain dataset by excluding zones used by large ISPs and hosting providers for reverse DNS delegation [53]. In simple terms, reverse DNS is the domain name that an Internet provider has delegated to an IP address. For example, for the IP address `173.53.80.48` the Internet provider has assigned the following reverse DNS delegation: `static-173-53-80-48.rcmdva.fios.verizon.r` This domain name can be retrieved by asking the PTR DNS record of the original IP.

Since malware execution may result in DNS PTR records to be created, we want to exclude the most frequently witnessed e2LDs in such reverse delegation. Therefore, we obtained a PTR scan of of all IPv4 from the Internet Systems Consortium (ISC), and we broke down all the e2LDs in this datasets according to the number of /24 and /16 network that can be seen. Our assumption here is that if the same e2LDs can be seen in several /16 and /24 networks, it must reflect a reverse DNS allocation conducted by an ISP or a hosting provider. We decide to pick the top 1% of the e2LDs for both /16 and /24 networks in our datasets, which reflects e2LDs that have been seen in more than ten /24 and /16 networks at the same time. This methodology identifies only 4,323 e2LDs—resulting in reverse pointer

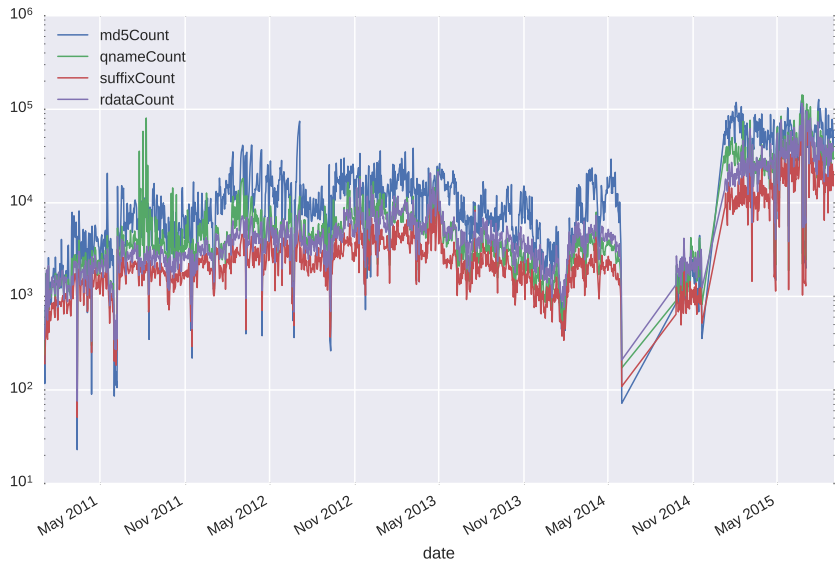


Figure 21: Number of malware samples, qnames, e2LDs, and IPs according to the execution time of the samples.

domains that are very likely associated with ISP or hosting networks. We use this list as the final filter, reducing the set of effective second level domains to 327,514 and 2,085,484 fully qualified domains.

Filtering Summary. The domain filtering phase reduced the initial candidate set of domains queried by malicious samples by over 95%. However, despite the significant reduction in e2LDs, 20M malware samples remain after filtering with at least one valid resolution.

Overall, this filtering was a very challenging and time-consuming process and the final result, as we will discuss later in the paper, still likely contains some benign domains with low popularity. However, we believe that our effort emphasizes two very important problems. First, the vast majority of DNS queries performed by malware are not malicious per se – and this may have a large impact on those approaches that populate domain blacklists based on the results of dynamic analysis sandboxes. Second, performing studies on very large datasets requires long periods—months in the case of this work—of manual work to tune filters and properly remove unwanted noise.

The final distribution of samples and domains over the four years of our dataset is

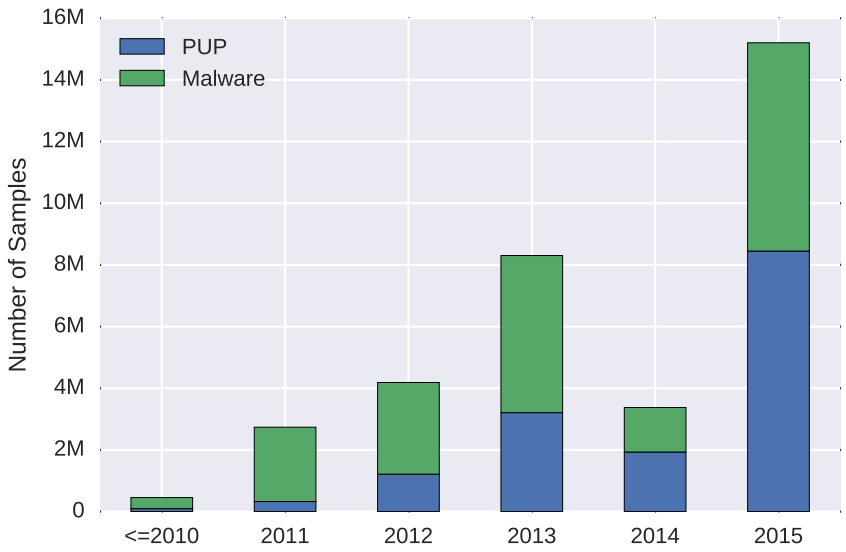


Figure 22: Malware and PUP samples over time. The drop in 2014 is due to a downtime of our largest feed of malware executions.

summarized in Figure 21. The drop in the second half of 2014 reflects a failure in our collection infrastructure for the largest feed of malware executions.

5.4 Classification

We perform 3 classifications on our dataset: grouping samples into families, classifying families as either malware or PUP, and assigning e2LDs to specific families.

Sample classification. We cluster and label all samples into known families using the AV labels in VirusTotal reports. While AV labels are known to be noisy [55, 166], we leverage AVClass [203], a recently released open-source tool for massive malware labeling. AVClass successfully removes noise from AV labels by addressing label normalization, generic token detection, and alias detection. The tool achieves F1 measures between 0.94 and 0.70 and it can process extremely large sets of VT reports—each containing AV scans of one sample by multiple AV engines. AVClass outputs for each sample the most likely family name and a confidence factor based on the agreement across engines.

PUP/Malware family classification. In addition to the family, AVClass also outputs for each sample whether it is PUP or malware by examining PUP-related keywords in the AV

labels. However, that classification is conservative as AV vendors often do not flag PUP samples as such. Thus, some samples in a family may be flagged as PUP and other samples in the same family as malware. To address this issue we have modified AVClass to output a classification for each family as PUP or malware, so that all samples in the family can be considered of the same class. Our modification counts for each family the number of samples flagged as malware and PUP. Then, it applies a plurality vote on the samples of a family to determine if the family is PUP or malware. We have contributed this modification to AVClass to be integrated in the tool.

We use our modified AVClass to automatically cluster and label 23.9M samples for which we have a VT report. As a comparison, the previous largest malware clustering/classification effort in the literature was the AVClass evaluation with 8.9 M samples [203]. Figure 22 shows the number of malware and PUP samples over time. The figure shows an increase of PUP samples over time, with PUP overtaking traditional malware since 2014. Kotzias et al. [142] observed the same trend but on a dataset two orders of magnitude smaller and from a single source, for which they could not discard source bias. Other work has also hinted on the prevalence of PUP. Thomas et al. [220] showed that ad-injectors affect 5% of unique daily IP addresses accessing Google. They also measured that Google's Safe Browsing generates three times as many detections for PUP as for malware [222].

AVClass identifies 17.7 K non-singleton families. Table 18 presents the top 10 families, which comprise 49% of the samples and are largely dominated by PUP. The largest malware families are `vobfus` (a Visual Basic worm [162]) and `virut` (a virus that appends its payload to other executable files [213]). Both families have self-replicating behavior that increases their polymorphism. The PUP families include adware that modifies advertisements or searches in the browser (`multiplug`, `loadmoney`, `hotbar`) [220] and a number of pay-per-install (PPI) programs (`softpulse`, `installerex`, `firseria`, `outbrowse`, `installcore`) [141, 222].

The 3,834 families with more than 10 samples comprise over 90% of all samples. Of

Table 18: Top 10 malware families by number of samples in our dataset. The FSeen column contains the first seen date of a family by VirusTotal.

Rank	Family	Samples	Type	e2LDs	FSeen
1	vobfus	2.8 M	Malware	741	11/09
2	multiplug	2.4 M	PUP	808	01/13
3	loadmoney	1.6 M	PUP	2,958	12/12
4	virut	1.4 M	Malware	40,705	03/08
5	softpulse	1.3 M	PUP	3,793	06/14
6	hotbar	1.1 M	PUP	306	08/10
7	installerex	847 K	PUP	155	12/11
8	firseria	795 K	PUP	3,138	07/12
9	outbrowse	771 K	PUP	52	04/13
10	installcore	661 K	PUP	1,118	09/11
Top 10		49%	-	15%	-

those families, 3,165 are malware and 669 PUP. While there are more malware families, the PUP families are larger with an average of 16 K samples per family compared to 3.5 K for malware families. This illustrates the highly polymorphic nature of PUP, which is not due to self-replication, but likely due to evasion of AV engines [142].

e2LD classification. We create a mapping from e2LD to the most likely family the e2LD belongs to. For this, we first create a mapping from e2LD to the number of samples of each family that have resolved that e2LD. Then, for e2LDs that have been resolved by at least 10 samples, we assign each e2LD to the family with most samples resolving it. e2LDs with less than 10 samples resolving them are left unclassified.

Table 19 presents the top 10 families ranked by the number of resolved e2LDs. Compared to the ranking by samples, this ranking is dominated by malware (8/10 families), which may indicate that PUP families have a more stable domain infrastructure and malware uses higher levels of domain polymorphism. The e2LDs from these 10 families correspond to 31% of all filtered e2LDs and Virut alone is responsible for 12% of them. Virut's domain infrastructure comprises a few dozens stable domains in the `.com` and `.pl` (Poland) TLDs, as well as a DGA. We study DGAs in Section 5.7. The popular `zbot/zeus` botnet is ranked third in Table 19, likely due to many different operators using the botkit.

Table 19: Top 10 malware families by number of filtered e2LDs that resolved to a valid IP address. The FSeen column contains the first seen date of a family by VirusTotal.

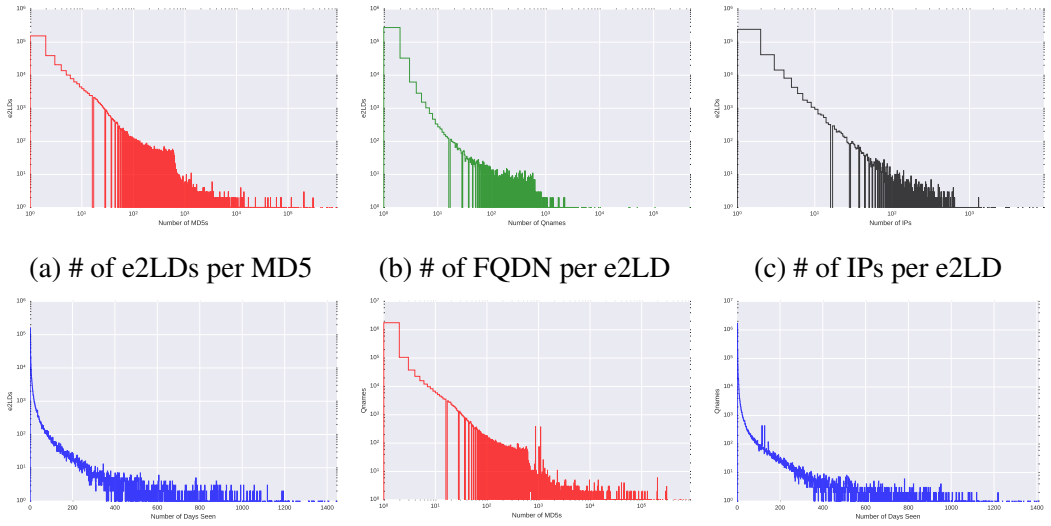
Rank	Family	e2LDs	Type	Samples	FSeen
1	virut	40,705	Malware	1.4 M	03/08
2	rodecap	17,382	Malware	11.8 K	05/09
3	zbot	12,959	Malware	163 K	01/08
4	tedroo	6,272	Malware	5 K	11/08
5	sality	4,964	Malware	463 K	12/08
6	upatre	4,658	Malware	503 K	09/13
7	fareit	4,217	Malware	61 K	10/11
8	softpulse	3,793	PUP	1.3 M	06/14
9	ircbot	3,635	Malware	28.5 K	05/06
10	firseria	3,138	PUP	795 K	07/12
Top 10		31%	-	17%	-

By combining the e2LD to family mapping and the family to PUP/malware mapping, we can mark e2LDs as belonging to malware or PUP. This classification identifies 36.5 K malware and 9.1 K PUP e2LDs. The remaining e2LDs are left unclassified due to less than 10 samples resolving them. This classification enables to study separately and compare properties of the PUP and malware network infrastructure (Section 5.5.2).

5.5 Malware Domain Analysis

Malware often engages in various network communications in an attempt to exfiltrate data, communicate with a command and control (C&C) server, or download additional illicit software. This communication often relies on DNS rather than static IP addresses to provide resiliency against IP blacklisting and ensure an overall agility for the malicious operation.

Therefore, we study the domains queried by malware to better understand the temporal DNS properties of their network communications. In Section 5.5.1, we evaluate domain names queried by malware during dynamic malware analysis. Our experiments show that malware frequently uses domain polymorphism that significantly limits the network policy and detection abilities of DNS blacklists. Then, in Section 5.5.2, we correlate those domain names with a large passive DNS dataset to identify whether we first collect the malware



(a) # of e2LDs per MD5 (b) # of FQDN per e2LD (c) # of IPs per e2LD

(d) # of Days Seen per e2LD (e) # of MD5s per FQDN (f) # of Days Seen per FQDN

Figure 23: Shows histograms of MD5 network traces broken down by various components.

sample or observe passive DNS activity for malware domains on the network. We find that a significant percentage of malware domains can be seen in passive DNS several weeks, in many cases even months, before the actual malware sample was dynamically analyzed by the security community.

5.5.1 Dynamic Malware Analysis

We start by analyzing domains collected from dynamic malware analysis. As noted in Section 5.2, we have a dataset of 26,853,732 malware samples collected since January 2011. From these samples, we collected 11,532,653 fully qualified domain names under 6,850,793 distinct effective second level domains. After extensive filtering, detailed in Section 5.3, we reduced this to 2,085,484 fully qualified domain names under 327,514 effective second level domains. In the following sections we study various properties of this set of domains.

5.5.1.1 Domain Polymorphism

Once a sample has been analyzed, the domain names used to facilitate malicious communication can be added to a DNS blacklist. Obviously, the effectiveness of these blacklists depends on how often different malware reuse the same domain names.

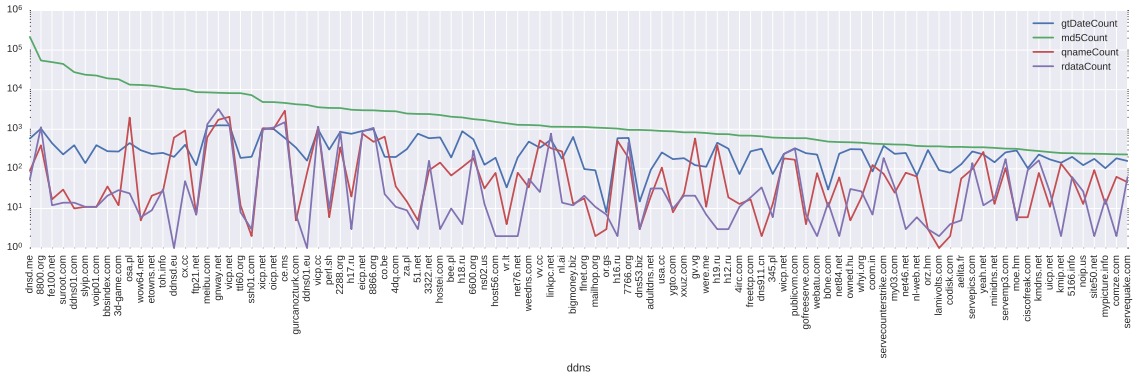


Figure 24: Top 100 most popular Dynamic DNS domains queried by malware samples.

The analysis of domains resolved by samples in our dataset shows that most malware samples appear to use different domains over time, as shown in Figure 23. In particular, Figure 23a shows that most MD5s resolve less than 10 unique e2LDs. Even more interesting, most of these e2LDs were seen only a single time across our five year collection period (Figure 23d), which means that they were only queried by a single malware sample. *This is an interesting result because it suggests that most domains are used only once by a single malware sample in our dataset.* If the domain is embedded in the binary and not downloaded from an external source, this can also cause samples in the same family to have different MD5s, even in absence of other polymorphism techniques. Furthermore, Figure 23b suggests that network evasion is being done predominantly on the e2LD since the majority of e2LDs have few child FQDNs. Further reinforcing this result, Figure 23e shows that FQDNs share an almost identical distribution to e2LDs.

These results suggest that *blacklisting malware domains observed during dynamic analysis does little to prevent future communication from newly discovered malware samples*. This result does not diminish the usefulness of collecting malware samples or performing malware analysis, but simply underline the limitation and reactive nature of relying on malware samples DNS queries for threat mitigation.

5.5.1.2 Dynamic DNS

Dynamic DNS allows nameservers to be automatically updated with frequently changing information. For example, users with dynamically assigned IP addresses commonly use dynamic DNS as a way of accessing their devices through an easy to remember domain name, which is updated as their IP address changes. There are numerous publicly available services that provide this functionality (e.g., [33,35]), and many of these services allow users to select a subdomain under a domain owned by the dynamic DNS provider, eliminating the need for the user to register a new domain name.

Due to its ability to provide rapid updates, dynamic DNS is also abused by malware authors to point domains at C&C servers or infected hosts. Furthermore, by using a domain provided by the dynamic DNS provider, the abuse cannot be blocked at the zone level without also blocking other legitimate users of the service. In fact, this has caused significant problems for past remediation efforts [13]. Additionally, Previous work [109] has shown that dynamic DNS domains are blocked at a higher rate (0.2%) than for all other web traffic (0.001%) as measured by data collected from Cisco Cloud Web Security (CWS). This suggests that there is a higher incidence of abuse for dynamic DNS domains. Unfortunately, no information was given about the scope or observation period of the data used to arrive at these numbers. In Section 5.5.1.2, we used our dataset spanning five years and 26.8M malware samples to perform a similar analysis. While we arrived at a similar conclusion that malware frequently makes use of dynamic DNS, our list of most frequently used dynamic DNS domains differed substantially this previous work. Since the popular dynamic DNS domains referenced in the previous work were a subset of those used in our study, this may indicate that the popularity of dynamic DNS domains for abuse varies over time.

Therefore, we decided to analyze which dynamic DNS providers are most frequently used by malware. In total, we found 718 known dynamic DNS e2LDs in our dataset from a list of dynamic DNS domains gathered from two public sources [32, 34]. Figure 24 reports the top 100 of these, sorted by the number of malware samples querying them. The

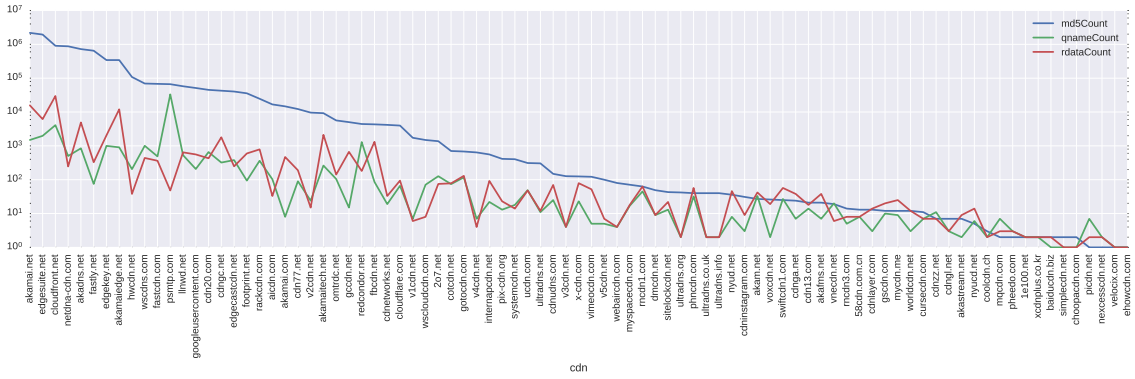


Figure 25: Complete list of *all* known CDN domains queried by malware samples.

most popular dynamic DNS domain, `dnsd.me` (owned by the dynamic DNS provider DNSdynamic [1]), was queried by 216,221 unique MD5s . This service is not only free, but it also offers unlimited registrations and an API for account management—making it very attractive for malware authors. Including `dnsd.me`, the top 50 dynamic DNS domains each have at least a thousand distinct malware samples that query them, and on average each of those domains has approximately 366 subdomains under it. In fact, we see that these top dynamic DNS domains account for 19,766 FQDNs. When looking at all 718 dynamic DNS domains, we see that they are queried by 8,675,449 distinct malware samples, which represents approximately 32% of all malware samples with DNS queries. Furthermore, these 718 domains account for 51,350 FQDNs. Thus, unlike most of the domains we discussed in Section 5.5.1.1, *dynamic DNS domains are commonly used across many malware samples and evasion is performed on the child label of the domain.*

5.5.1.3 Content Delivery Networks

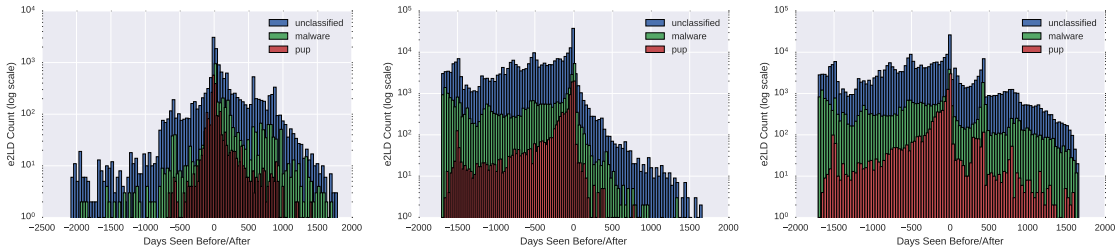
Content Delivery Networks (CDNs) are frequently used to serve content from multiple, geographically distributed, data centers to provide increased performance and availability. By taking the client location into account, CDN providers are able to serve up content from the nearest data center, improving network performance. Most providers are still able to offer performance benefits even when location information is unavailable due to faster

connections and high-end data centers. Additionally, serving content from multiple data centers helps obviate content outages by providing network redundancy. It is no surprise, given their benefits, that CDNs are widely used on the Internet.

In this section, we study how malware uses CDNs by studying domains collected from dynamic malware analysis. Figure 25 shows a plot of all CDN domains, sorted by how many unique malware samples queried them in our dataset. The first notable feature of this plot is the discrepancy between the most and least popular CDNs. The top five most queried CDN domains include `akamai.net`, `edgesuite.net`, `cloudfront.net`, `netdna-cdn.com`, and `akadns.net`. This list includes some of the largest CDNs and is not dissimilar from what a benign network application might be seen querying. Another interesting insight from Figure 25 is the number of malware samples using CDNs. The `akamai.net` domain alone is queried by 2,183,352 distinct malware samples and has 1,492 unique child labels under it. *The large number of child labels combined with potentially benign usage allows malicious content hosted in a CDN to effectively hide in plain site.*

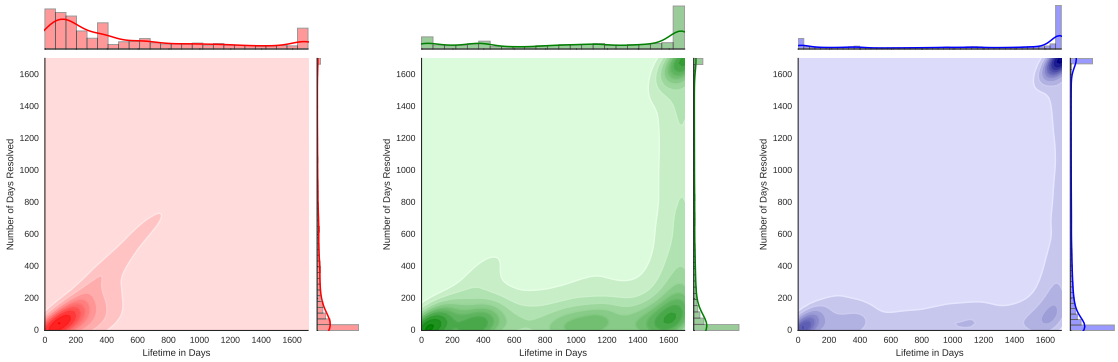
5.5.2 Passive DNS and Blacklists Analysis

In the previous sections, we analyzed the domains collected from network traces observed during dynamic analysis. In this section, we correlate these domains with three other data sources: (1) a passive DNS dataset provided by a large ISP in the United States, (2) a number of public DNS based blacklists, and (3) a set of domain expiration events. This allows us to study the lag between when a domain is discovered through dynamic malware analysis, or listed on a blacklist, and when it is first resolved in passive DNS. This allows us to better understand the implications of relying on dynamic malware analysis of public blacklists for early detection systems.



(a) PBL (b) Passive DNS (c) Expired

Figure 26: Time difference between when a domain was first seen in passive DNS, public blacklists, or an expired domain list rather than through dynamic malware analysis.



(a) PUP (b) Malware (c) Unclassified

Figure 27: Joint distribution of domain lifetime and resolution frequency observed in passive DNS for PUP, Malware, and Unclassified domains.

5.5.2.1 First Appearance

We start our analysis by evaluating efficacy of public blacklists at identifying malware domains. This provides an interesting perspective because domains on these lists have already been flagged as abusive by manual experts or dedicated services. The result of this analysis is plotted in Figure 26a, separated by the type of sample. As we explained in Section 5.4, we classify domains in our malware analysis traces as belonging to a malware family, PUP, or an unclassified category—which comprises e2LDs resolved by less than 10 samples. This separation allow us to provide insights into potential differences between these three classes of malicious software.

The figure shows that many domains were added to public blacklists only after we observed them in dynamic malware analysis traces. In particular, only 30% of the entries

were added to blacklists before the domain was observed in our dynamic analysis dataset, while 20% of them were reported with a delay of over 500 days. *This result suggests that such delays could be largely reduced by relying on malware analysis to populate domains blacklists—possibly after applying a cleaning methodology like the one we described in Section 5.3*. While it may seem reasonable to attribute this delay to the selection of blacklists used in this study, this result is consistent with previous work by Kührer et al [143] where domains were seen in passive DNS on average 384 days before appearing on a blacklist. Therefore, it is unlikely that the addition of other blacklists would profoundly affect this result. Additionally, reputation systems [43, 62] that rely on passive DNS have also demonstrated the ability to identify new threats more quickly than public blacklists. Furthermore, the observed delay between appearing in passive DNS and on public blacklists also lends credence to the idea of proactively detecting and blocking abuse at the time of domain registration as proposed by Hao et al. [113].

Next, we compare the date when we first observed a malware domain resolve in passive DNS with the date when the same domain was first observed in a dynamic malware analysis trace. By computing the difference between these two dates, we can determine how quickly new malware threats are discovered and analyzed by the security community. Figure 26b shows whether a malware domain was first seen in passive DNS or in a network trace derived by the dynamic analysis of a malware sample. Points less than zero on the x-axis indicate that a domain was first seen in passive DNS, and points greater than zero mean that the malware discovery occurred before the first observed network resolution in passive DNS.

The figure shows that the PUP-related domains are active an average of 192 days before we get to dynamically analyze the corresponding samples. This may be expected, as PUP relies on infrastructure that is more stable and long-running. However, we can see that popular malware families also follow a similar but less extreme pattern. This result is more surprising because for most of these domains the difference is very significant—with discovery delays reaching 623 days on average. Lastly, the domain names associated with

the unclassified category follow the most interesting distribution. While many appear first in the passive DNS traces (left side of the Figure 26b), this category completely dominates the right tail of the graph—representing domains that were seen in passive DNS only months after we observed them in our sandboxes.

Overall, by combining the three classes together, we discovered that 302,953 malware domains were active at least two weeks—in some cases many months—before the corresponding malware samples were analyzed. *Therefore, while we previously showed that dynamic analysis systems could be used to improve current blacklists, our results also show that blacklists built from dynamic malware analysis will still be unaware of potential threats for several weeks or even months.*

The surprising nature of this result prompted us to perform additional analysis. Thus, we used our dataset of historic domain expirations to verify that a given domain was not used in the wild before expiring and being re-registered for malicious purposes. Figure 26c shows a similar pattern as the one in Figure 26a. The large peak around zero is likely a result of changes made to the domain infrastructure for unpopular or unused expiring domains. Such changes would likely result in DNS traffic to a parked or suspended page during the registrar's expiration grace period. Despite our extensive filtering efforts, the left tail in the graph remains significant. This can be partially explained by malware relying on benign infrastructure, such as dynamic DNS and CDN providers already mentioned in Section 5.5.1. Another possible explanation is that the long tail is an artifact of a long setup phase for malware before it is released into the wild. During this phase, malware authors may age the domain and point it to benign infrastructure to build up positive reputation to help evade detection later. However, in the case of expired domains, this step is unnecessary because the domain will inherit the residual trust associated with the domain [153]—eliminating the need for a long aging phase. As shown in Figure 26c, we *still* see a long delay between last expiration of a domain and first discovery of an associated malware sample. Thus,

one explanation could be that there is a long delay between a domain expiration and re-registration. In fact, this very behavior was seen in a study of spam related domain names by Hao et al. [114]. Their research showed that many registrations came hundreds of days after expiration, but abuse was observed less than three months after re-registration in most cases. Since we removed spam related malware samples in our extensive filtering step, this could suggest that the same behavior employed by spammers may also be used by other malware authors. However, it is also possible that domains are registered quickly after expiration and immediately used for abuse. Another potential explanation may be that our feeds are not the first to see new samples. However, since we are also considering first seen date from VirusTotal, it seems unlikely that the addition of other feeds that would dramatically reduce the first seen date for the malware samples in our dataset.

5.5.2.2 Domain Lifetime

Finally, we look at the lifetime of the malware domains using Figure 27, which shows a joint density distribution between the number of days a domain was resolved and the lifetime of that domain in days for PUPs, malware, and unclassified samples. We define the "lifetime" as the difference between the first and last seen dates for each of these domains in passive DNS.

The joint distribution makes it easy to infer not only how often a domain resolves but also for how long. In Figures 27b and 27c, we notice that there are three hotspots that correspond to the most prevalent resolution behaviors for domains in malware and unclassified malicious software. In the bottom left, we see that there are a large number of domains that are short lived and rarely resolved. A second hotspot, in the top right corner of both figures, corresponds to domains with the exact opposite behavior—long lived and frequently resolved. These domains were regularly queried for the entire duration of our experiments. Finally, in the bottom right of both figures, we observe a third pocket of domains that have a long lifetime but infrequent resolution. In this case, we observed only a

few queries that were often years apart.

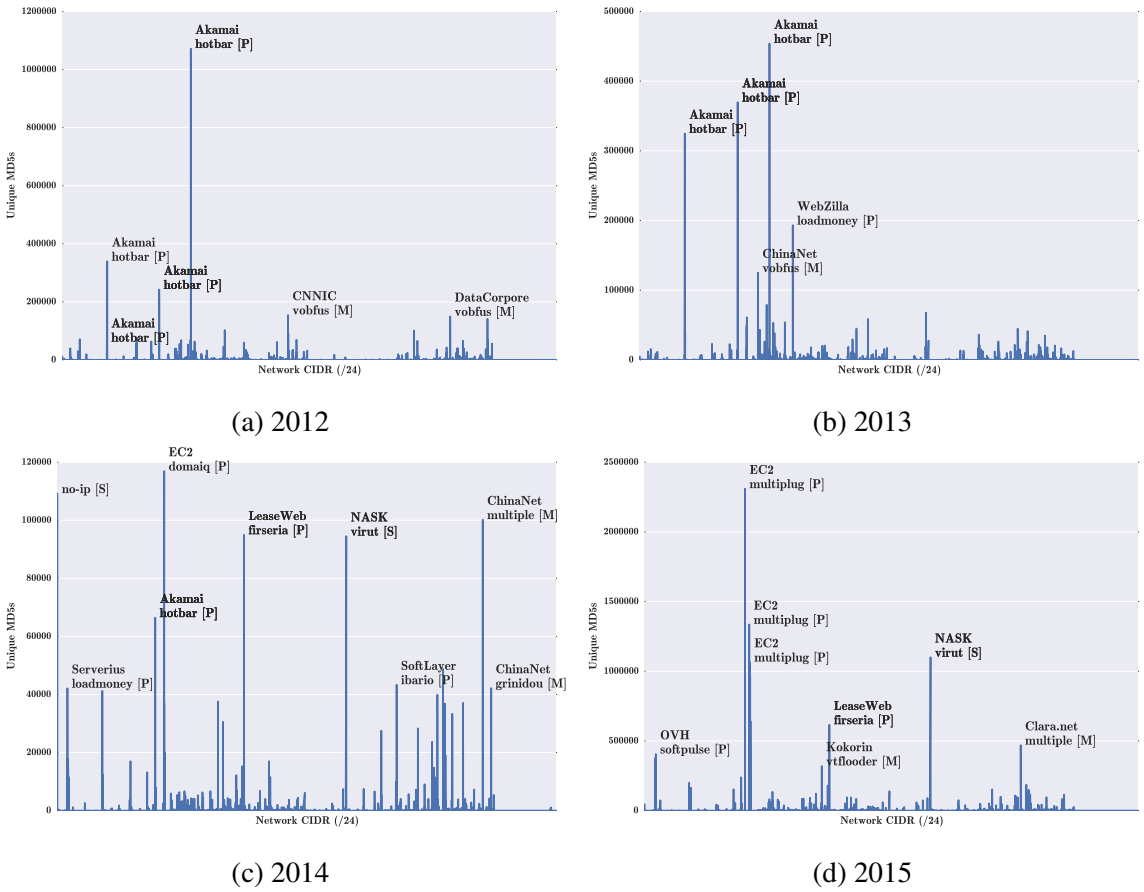
If we focus our attention on Figure 27a, which shows the lifetime patterns for PUP domains, we observe a completely different distribution. This is the consequence of two phenomena. First, we have seen the prevalence of PUP domains rise over the last two or three years, and this fact justifies the bounding of the joint distributions in the $[1,000, 1,000]$ region. The second reason has to do with the seemingly intense and continuing resolution of PUP related domains—which manifests as a higher density along the diagonal. We believe that this is a result of organizations failing to block PUP domains and end-point security engines that do not manage to remediate PUP infections. As Figure 27a shows, this gives PUPs a significantly different DNS resolution profile. On the other hand, the unclassified domains shown in Figure 27c follow a very similar pattern to the malware domains shown in Figure 27b. This likely indicates that most unclassified domains are very likely malware domains.

Summarizing, Figure 27 makes it clear that the all three types of domains frequently have long domain lifetimes, and many of those domains are frequently looked up. Since we showed that most domains were only resolved by a single sample in Section 5.5.1.1, this suggests that many samples remain active on the Internet for extended periods of time.

5.6 Infrastructure Analysis

In this section, we analyze the hosting infrastructure for the domains resolved by the malware samples in our dataset. In particular, we want to investigate whether certain IP ranges appear more often the others, what are the reasons behind this choice, and how the global infrastructure picture evolved over time.

Figure 28 shows a histogram of the number of samples with domains (after filtering) resolving into a given /24 subnet, for each year between 2012 and 2015. In each plot, we observe spikes indicating that certain subnets are resolved by a very large number of samples during that year, from hundreds of thousands of samples in 2013 and 2014, up to peaks of



(a) 2012 (b) 2013

(c) 2014 (d) 2015

Figure 28: Histograms of number of samples resolving domains that point to /24 subnets. Spikes are annotated with the owner of the IP range, the family that contacted it, and a letter indicating whether IPs are associated with malware(M), PUP(P), or a sinkhole (S).

few million samples in 2012 and 2015. Some of the spikes can be observed on multiple years e.g., `66.150.14.0/24` (Akamai) in 2012–2013 and `148.81.111.0/24` (NASK Polish CERT) in 2014–2015, while the majority appears in a single year.

We can assign spikes to specific families by analyzing the e2LDs that resolved to those ranges and use the mapping of e2LD to family produced by our classification in Section 5.4. This analysis reveals that there are three different reasons behind these spikes.

The largest group corresponds to spikes caused by specific malware families reusing the same subnet (sometimes even the exact same IP address) for long periods of time. Those families correspond to some of the Top-10 families by number of samples in Table 18. The majority of these spikes are due to PUP families (`hotbar`, `domaiq`, `firseria`,

multiplug) although we also observe some spikes due to polymorphic malware like vobfus. For example, three of the top four spikes in 2015 correspond to Amazon EC2 ranges and are all due to e2LDs belonging to the multiplug PUP family, which seems to have migrated its hosting infrastructure to EC2 in 2015. Similarly, the top spike in both 2012 and 2013 is caused by the Hotbar PUP family. This family used the Akamai CDN in 2012–2014 to host its infrastructure and therefore caused multiple spikes in different Akamai IP ranges. While we have observed a large number of benign domains resolving to the Akamai ranges, after our filtering in Section 5.3 it was simple to manually recognize the Akamai spikes in 2012–2014 and associate them to the Hotbar family.

The second group of spikes corresponds to sinkholes used to redirect resolutions of malicious domains after intervention. The most visible spike in this category is 148.81.111.0/24 in 2014–2015, which is due to sinkhole.cert.pl, used by NASK since 2014 to sink resolutions of the Polish domains used by the Virut botnet. Our dataset contains 123 Virut e2LDs resolving to this sinkhole being contacted by over 1M Virut samples in 2015. Another example of sinkhole-related spike is 0.0.0.0/24 in 2014, which is caused by a Microsoft-lead intervention on domains used by the no-ip dynamic DNS provider [13].

The third group of spikes is due to multiple, rather than a single, malware family. These indicate hotbeds of abuse and appear to keep changing over time. They may correspond to hosting providers that, over a certain time frame, had a more open policy on acceptable behavior. One such spike is on the right of the 2015 plot and corresponds to Clara.Net (195.22.26.0/24), a Portuguese hosting provider that hosted many domains associated to the sality, wapomi, ramnit, and techsnab malware families. Another example is the on the right side of the 2014 plot and corresponds to ChinaNet (218.92.221.0/24), where we observe domains, among others, of the frethog and karnos malware families.

Regarding hosting providers, the top spikes in 2012 and 2013 correspond to Akamai (due to hotbar) and ChinaNet (vobfus). There is also one spike in 2013 due to loadmoney

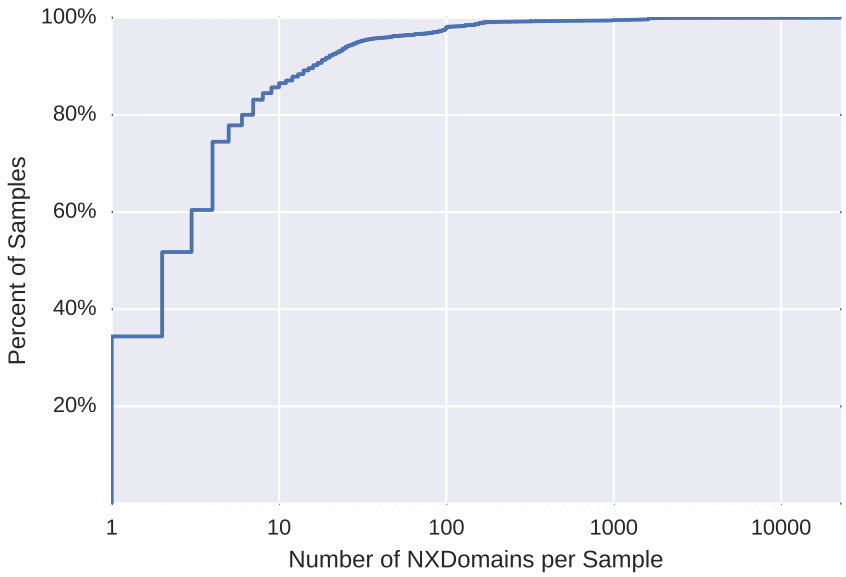


Figure 29: Cumulative distribution function (CDF) for the number of NXDomains seen in malware samples in our datasets.

in the WebZilla cloud hosting service. As an interesting observation, starting from 2014 most spikes occur on IP ranges belonging to cloud hosting providers—most notably EC2, LeaseWeb, OVH, Serverius, and SoftLayer. One exception is a 2015 spike due to the `vtflooder` malware that resolved to `91.223.216.0/24`, which is registered to a private Ukranian person that uses a Gmail abuse email address.

Our analysis shows that the large spikes are dominated by PUP families and can last for multiple years indicating that PUP utilize seemingly stable IP infrastructure. This may indicate that popular cloud hosting providers like Amazon EC2, LeaseWeb, or OVH, and CDNs like Akamai, where PUP spikes happen, may not have the same policies towards banning PUP that they use for malware.

5.7 DGA Malware

In the previous section, we provided an extensive analysis of the evolution of malware network infrastructure based on successful DNS resolutions. We now focus on a different aspect of malware behavior: the presence and impact on our dataset of domain name

generation algorithms (DGAs).

Since the vast majority of DGA-related queries do not resolve to a valid IP address, to perform this analysis we first need to reintroduce in our dataset the failed (NXDomain) resolutions we filtered out in Section 5.3. Since 2011, over 12.5 million malware samples in our dataset produced at least one NXDomain during their execution. The cumulative distribution function (CDF) in Figure 29 shows how half of the malware executions have less than two NXDomain resolutions and only 950,644 have over five.

To identify DGA-generated domains, we check if the domains in our malware executions dataset appear in the DGArchive [31], which comprises 50M domains generated by the DGAs of 66 malware families. Table 20 summarizes the overlapping of the DGArchive domains with both failed and successfully resolved domains in our malware executions dataset. For each family, it first shows how many e2LDs in the DGArchive are observed among the 6.8 M domains in our dataset before any filtering (i.e., including NXDomain queries). Then, it shows how many e2LDs from the DGArchive are observed among the 327 K domains remaining after all filtering.

According to the DGArchive, at least 44% (3 M) of all the e2LDs observed in our malware executions, regardless whether they successfully resolved or not, were generated by DGAs. This percentage is a lower bound since the DGArchive likely misses some DGA families, and also some variants that modify the DGA algorithm or its seed. After filtering, at least 17% of remaining e2LDs come from DGAs.

Our malware executions contain DGA e2LDs for 42 out of 66 (64%) families in the DGArchive. This number highlights the large coverage of our dataset. Of those 42 DGArchive families, 4 are variants of another family (e.g., pykspa and pykspa2), which we group into the 38 families in Table 20. The large majority of DGA domains (82%) come from `virut`, followed by `pykspa` (6%) and `necurs` (4%). After filtering, successfully resolved `virut` DGA domains correspond to 12% of all unfiltered domains, followed by `zbot-gameover` (4%), and `ramnit` (0.5%). While `virut` is the most common DGA

family in our dataset (and is still very active despite the 2014 takedown), if we normalize by number of samples in the family, we observe that `virut` resolves 1.8 e2LDs per sample, well below the most aggressive families: `emotet` (82 e2LDs/sample), `murofet` (68), and `cryptolocker` (53).

The table shows that only 1.8% of DGA domains successfully resolved. It also emphasizes different DGA behaviors by the malware families. For example, while we observe over 110K domain names queried by `necurs`, only one of them was active during our analysis. This indicates lack of pressure by defenders so the family can keep using the same domain. On the contrary, in other families like `gameover` roughly 30% of the queried domains resolved to a registered IP address.

The 24 families in the DGArchive that we do not observe are likely due to two reasons. First, some malware use the DGA as a backup option in the malware's communication strategy, i.e., only used if the primary C&C domains fail. This may not happen during the few minutes the samples are executed in the sandbox. Second, for some malware like `TDSS/TDL4`, the DGA is only used in specific components, i.e., monetization through ad abuse. Thus, we may not see DGA domains in our traces because the malware samples have not run long enough to be monetized.

Finally, in our dataset we also observe five candidate DGA families, not included in the DGArchive, which perform large numbers of NXDomain queries and have random-looking domains (`fosniw`, `shiz`, `softpulse`, `upatre`, `wapomi`).

5.8 Spam Related Malware

A common way to monetize malware is as distributed mechanism for sending spam. As discussed in Section 5.3, we excluded spam related malware from our analysis because it is outside the scope of this work, and we did not wish to influence our analysis with network activity that is characteristic of typical spam behavior. Since there is already plenty of work that studies spam in great detail, we provide the following discussion only to provide more

insight into the spam related samples we excluded from our analysis.

Prior to filtering, we identified a number of malware samples in our dataset that appeared to be involved in spam behavior. After gathering a set of domains used as MX records or that were likely to be associated with mail related activity, we built a collection of 405,742 malware samples that contacted these domains. This collection represents malware samples that were potentially involved in spamming activity. As seen in Figure 30, we noticed that these samples could be seen contacting hundreds to thousands of unique domains. This observation can also be seen in Table 21 that shows the ratio of MX lookups per malware sample. With the exception of Sality, the top 25 malware families queried at least 40 MX records. Since most malware we studied contacted less than 10 domains, this is an interesting characteristic of many of these samples.

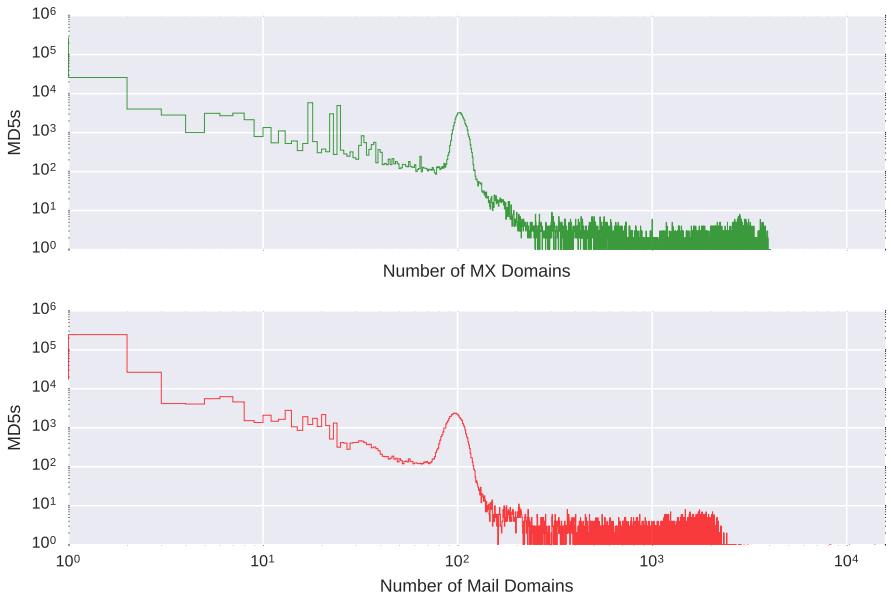


Figure 30: Shows a histogram of the number of MD5s associated with each spam related domain in our filtering set.

Figure 30 shows the number MX lookups and mail related DNS queries for each potentially spam related malware sample. Notice that there is an interesting peak around 100 domains. After associating 5,239 samples with this spike, we were able to obtain VirusTotal reports for 909 (17.4%) of these samples; 892 (98.1%) were instances of Mydoom [239]

malware—a compute worm first sighted in January 2004 and used by e-mail spammers to send mail from infected hosts. As noted in Section 5.2, our earliest malware feeds start in January 2011, which is a full seven years after the first sighting of the Mydoom e-mail worm, and despite being known for over a decade, we still saw active Mydoom variants in our malware dataset as recently as August 2015. The long lifetime of this particular malware family is interesting because it suggests that even older malware is effective for spam related activity.

5.9 Summary

After carefully filtering 26.8 million network traces obtained from dynamic malware execution, we are able to make several observations about the characteristics and temporal network properties of malware domains. First, we show that dynamic analysis traces should be carefully curated because they often contain a great deal of noise. To help with this challenge, we detail a rigorous methodology that analysts can use to remove potential noise from such traces. Next, potentially unwanted programs (PUPs) are not only on the rise, but surprisingly, they utilize seemingly stable IP infrastructure. In fact, we show that several hundred thousands PUP samples use the same network infrastructure over an entire year. Finally, our analysis shows that malware appears to add marginal detection benefits when trying to build early warning systems based on its network communication. We discovered that 302,953 malware domains were active at least two weeks—in some cases many months—before the corresponding malware samples were dynamically analyzed. This means that malware domain blacklists have limited detection value as malware tends to rapidly churn through domain names—yielding a very high rate of domain-level polymorphism.

Table 20: DGA e2LD in the DGArchive [31] resolved in the malware executions in our dataset.

#	Family	Before Filtering	After Filtering
1	virut	2,477,628	40,452
2	pykspa	189,644	180
3	necurs	110,092	1
4	suppobox	72,476	4,677
5	tinba	52,463	682
6	gameover	24,325	7,083
7	emotet	23,500	96
8	pushdo	13,170	17
9	ranbyus	12,922	7
10	nymaim	12,490	148
11	simda	12,348	590
12	murofet	9,295	20
13	qakbot	4,130	119
14	ramnit	3,560	418
15	cryptolocker	2,912	89
16	conficker	1,710	465
17	sisron	1,394	1
18	oderoor	622	3
19	matsnu	525	130
20	dircrypt	510	53
21	tempedreve	204	20
22	banjori	200	1
23	feodo	192	13
24	urlzone	77	18
25	tsifiri	59	58
26	torpig	53	2
27	ramdo	49	27
28	gspy	49	0
29	bamital	48	2
30	bedep	44	5
31	hesperbot	37	2
32	fobber	31	2
33	gozi	24	8
34	bobax	23	0
35	proslikefan	12	1
36	darkshell	10	3
37	redyms	2	2
38	xxhex	1	1
All		3,026,831	55,396

Table 21: Top 25 spam families (filtered) ranked by number of MX lookups.

Rank	Family	Samples	MX	Ratio
1	mydoom	82.0 K	6.0 M	72.8
2	hlux	1.7 K	3.5 M	2047.2
3	zbot	1.2 K	1.1 M	928.2
4	fareit	953	403.3 K	423.2
5	kelihos	446	388.0 K	870.1
6	winlock	171	373.3 K	2183.0
7	upatre	58	100.6 K	1734.0
8	zusy	54	92.0 K	1702.5
9	sality	23.6 K	86.2 K	3.7
10	tofsee	523	81.6 K	156.1
11	slym	38	63.7 K	1662.2
12	agentb	58	60.6 K	1045.2
13	tedroo	1.3 K	57.0 K	44.6
14	glupteba	21	45.1 K	2149.0
15	mikey	35	44.7 K	1276.2
16	bredolab	88	42.4 K	481.7
17	vblv	18	38.1 K	2115.8
18	yakes	85	35.9 K	422.4
19	tinba	13	34.8 K	2677.8
20	waledac	44	32.2 K	732.9
21	pwszbot	15	29.9 K	1992.4
22	ceeinject	19	26.0 K	1367.5
23	dorifel	195	24.8 K	127.5
24	zboter	10	23.5 K	2355.2
25	staser	2	21.6 K	10786.0
Top 25		112.7 K	12.8 M	113.3

CHAPTER VI

RETROSPECTION ON STUDIES

Each of the studies presented in this thesis examines a particular threat over some fixed snapshot in time. However, beyond simply analyzing security issues of the day, each study also raises interesting research questions that extend beyond the temporal window of the original work. This section will explore some of these open research questions and discuss how they provide interesting research directions for future work. It will also attempt provide new hypotheses, based on extended reflection, for interesting phenomena discovered in those studies.

6.1 Mobile Malware

In Chapter 3, we presented an empirical analysis of mobile malware using DNS traffic from a major cellular provider in the United States. Counter to the prevailing wisdom of the day, this work found very little network activity to known mobile malware indicators of compromise— suggesting low infection rates for mobile devices on the cellular network. In this section, we discuss some interesting research directions based on those findings and evaluate how those initial results have held up over time.

6.1.1 Tainted Infrastructure

Despite limited evidence of mobile malware communication from mobile devices, this study observed a large volume of requests to tainted network infrastructure. This tainted infrastructure was often associated with other known forms of abuse such as phishing, drive-by-downloads, or traditional malware that did not necessarily target mobile devices. This raises an interesting question of why mobile devices frequently reach out to this infrastructure.

One hypothesis is that users are frequently using their mobile devices for activities previously done on a traditional computing device. For example, users frequently use mobile devices to view and respond to e-mails as well as perform basic web browsing—tasks previously done on traditional computing devices. Consequently, users may frequently click on unsafe links in e-mails or navigate to unsafe websites on mobile devices instead of a laptop or desktop. The result is mobile devices are visiting infrastructure historically associated with abuse.

While some abuse targeting traditional computing devices may not affect mobile devices, some traditional types of abuse may be more effective against mobile platforms. For example, existing research has shown that constraints imposed by mobile devices may make mobile users more susceptible to phishing attacks [42,94]. Beyond being more susceptible to certain types of attacks, mobile users are not immune to existing threats. In 2011, a vulnerability in the iOS PDF viewer allowed iOS devices to be rooted from the web browser [207], and while this was marketed as an easy way for users to jailbreak their devices, it demonstrates that mobile users should still be cautious about the links and websites they visit.

As mobile device usage continues to rise, Internet adversaries may begin to deploy targeted exploits for the device types visiting tainted infrastructure. Thus, tainted infrastructure previously innocuous to mobile devices may represent a real security threat. As a result, an interesting line of research would be to further study what types of exploits exist at the tainted infrastructure visited by mobile devices—ascertaining whether we see a rise in the number of targeted exploits against mobile devices.

6.1.2 Mobile Application Markets

At the time of our study, the number of mobile devices was increasing along with the number of malware samples targeting mobile platforms. However, we still found very little evidence of mobile devices actively infected with malware. One potential explanation for these low infection rates is the use of different security paradigms on mobile devices. For example, the

major mobile device platforms such as the iOS, Android, Windows Phone, and Blackberry each offered their own first party application markets—centralized repositories of vetted applications [103,133,240,243]. While software repositories have existed on other platforms in the form of package management systems [242], these first party application markets made managed application distribution available and accessible to a much larger group of end users.

These application marketplaces have grown over the years and offer users a seemingly endless supply of mobile applications [119,131]. The sheer number of applications available in these application markets can make it difficult for users to discover new applications—including malicious applications. Even mobile malware researchers have struggled with finding malicious applications in these markets, and consequently, previous work has built automated tools to efficiently scan mobile application markets for likely abusive applications [69,254]. Given the challenges encountered by researchers actively looking for malicious applications in mobile markets, discoverability in large mobile application markets may play a factor in the low malware infection rates seen in the aforementioned study.

To aid in discoverability, many mobile marketplaces offer the ability to link directly to the applications they host [104,132]. These links can be shared anywhere a normal uniform resource indicator (URI) can be shared—including websites, social media, or even e-mail. Such links make it easy for an application developer to direct users to their marketplace listing without requiring the user to search through the marketplace. However, these links can also be used by malware authors to direct users to abusive applications. Phishing e-mails could be sent to end users with market links for re-packaged versions of legitimate apps that steal user credentials, exfiltrate user data, or worse. Spam or advertisements could also be used to scare users into downloading fake versions of security tools—replicating a scam seen in abuse against traditional computing devices.

Given the challenges around discoverability in mobile marketplaces, it would be interesting to see how mobile market links are used and potentially abused by abusive applications. A study that surveys where mobile market links are most frequently observed, how often they are used for abuse, and what features of such links potentially indicate abuse could help facilitate a better understanding of the mobile malware ecosystem and improve overall security even further.

6.1.3 Infection Rate Changes

The original study presented in Chapter 3 presented a snapshot of abuse in 2012. A natural question is whether the initial results reported in this study have changed over time. Fortunately, several industry groups have performed independent studies since the publication of our original study.

In 2015, Verizon performed a study that measured the number of mobile devices infected with mobile malware on their network, and the results of this study were published in their Data Breach Investigations Report (DBIR) [16]. Ultimately, the report observed infection rates very similar to the results in our study—finding that only about 0.03% of smartphones, out of tens of millions of mobile devices, per week were infected with malware applications.

Google also publishes a yearly Android Security report that includes measurements of of global infection rates of potentially harmful applications (PHA)—which Google defines as apps that could potentially put users, user data, or devices at risk. The latest report [38] indicated that at most 0.12% of applications installed from the Google Play marketplace were considered potentially harmful throughout 2017. The percentage rose to 0.92% when looking at applications not installed through Google Play—suggesting that the majority of abusive applications are sourced outside the first party marketplace. These numbers are especially interesting given Google's global device visibility and access to system level installation information.

Ultimately, despite the increase in malware samples [161] available for mobile devices,

the actual number of infected devices still remains at less than 1% for the largest mobile platform in 2017. While much has changed in the mobile ecosystem over the years, mobile malware infection rates still represent a real but very small threat—especially for users installing applications exclusively through first party markets like the Play Store.

6.2 Residual Trust

In Chapter 4, we presented an empirical study of residual trust abuse and demonstrated that it has become increasingly common in recent years. While our study focused on domain expirations, these are not the only source of domain ownership changes. In this section, we will discuss some interesting extensions of this work as well as some future challenges in this area.

6.2.1 Beyond Expirations

One way that domains frequently change ownership is through domain expirations. However, this is not the only source of domain ownership changes. Sometimes domains are bought and sold just like investments. In fact, there is an entire industry around domain name speculation [241], where certain domains are bought with the belief that their worth will only grow over time.

Sometimes this speculation is done at the expense of existing brands and trademarks and is referred to as cybersquatting. The squatter may then try and sell the domain to an organization that owns the brand or trademark [163], but more frequently cybersquatted domains are associated with various types of domain abuse such as typosquatting [39], bitsquatting [78], or combosquatting [139]. Domains used in this manner may sometimes be reclaimed by a trademark or brand owner through the Uniform Domain Displute Resolution Policy (UDRP) [128] set forth by ICANN, and many domain speculators avoid cybersquatted domains for this reason.

Instead, many domain speculators will focus on domains with characteristics like shorter length, common words, and older top level domains that have a perceived higher value [121].

These domains are frequently registered with the sole intent to resell or license their use to an interested entity. Once a domain has been registered, there are numerous secondary markets that facilitate the selling or trading of domains between two parties. Similar to second hand markets like eBay, these sites typically allow sellers to make domains available via a fixed price sale, best offer, or auction. Ultimately, the seller will transfer domain ownership to the purchaser of the domain resulting in a change of ownership without the domain expiring. In some cases, this may even be facilitated by a third party escrow service [86].

This second hand ecosystem of domain ownership changes represents an interesting area for future research. Domains obtained via these services likely transfer a significant amount of positive residual trust to new owners, and therefore, they may be of particular interest to Internet adversaries.

6.2.2 General Data Protection Regulation

In April of 2016, the European Union (EU) adopted a new data protection and privacy regulation named The General Data Protection Regulation (GDPR) [100]. This legislation seeks to control the export of personal data outside of the EU and give citizens control over how that data is used. Enforcement of new GDPR regulations will go into effect on May 25, 2018. While there are many exacting legal requirements, compliance generally requires companies to carefully handle user data and provide EU citizens with ways to monitor, control, and delete their data if desired. For technology companies, this means that user data collected about EU citizens, online or otherwise, will be subject to GDPR regulations.

One source of data that will be affected by new GDPR regulations is domain name WHOIS information collected by Internet registrars. As discussed in our study on residual trust, WHOIS data can be used to identify changes in domain ownership, but it can often be difficult to collect at scale. New GDPR regulations will affect the information collected by many Internet registrars and may further compound the challenges of working with WHOIS data. As a result, ICANN currently has several initiatives [127] working to address

the challenges GDPR will impose on existing registrars and their ability to fulfill existing contractual obligations. It is unclear what the future holds for WHOIS data [177], and ultimately, GDPR regulations could result in the WHOIS database being dissolved, requiring gated access, or disallowing bulk access to records.

The GDPR and uncertainty around the future of WHOIS reflect the public's growing frustration with how sensitive personal data is handled. As a result, security researchers may find it more difficult to obtain this information and might need to rely on other techniques for identifying down ownership changes. This further highlights the value offered by algorithms like Alembic, as discussed in Section 4.4.2.2.

6.3 Traditional Malware

In Chapter 5, we presented a longitudinal study of malware over a period of five years—allowing us to identify trends in malware behavior over time. In addition to confirming the results of some smaller scale studies, we discovered several new findings that raised interesting new questions. In this section, we dig a little deeper into some of those findings and suggest some potential directions for future research.

6.3.1 PUP Behavior

One interesting phenomenon that we identified in our study was the rise of Potentially Unwanted Programs (PUP)—confirming the results of previous work done on a smaller scale [141]. In particular, we saw that the number of distinct PUP samples has grown at a faster rate than malware samples in recent years. This is interesting because we show that PUP samples appear to exhibit network properties that are distinctly different from that of malware, as clearly seen in Figure 27.

In particular, we observed that PUPs frequently relied on the same network infrastructure over extended periods of time. However, despite their reliance on extremely stable infrastructure, PUPs did not appear to be remediated more quickly than traditional malware. One potential hypothesis for this behavior is that PUPs rely on infrastructure that may be more

difficult to block because it is associated with otherwise benign uses, and Figure 28 shows that a number of PUPs rely on infrastructure hosted on popular CDNs or cloud providers—a fact which would be consistent with this hypothesis. Remediating access to PUP infrastructure would require an organization to block infrastructure not exclusively associated with abuse, but the false positives incurred would likely prohibit most organizations from adopting such a policy.

We also observed that communication with the same infrastructure happened on a very consistent, periodic basis. Thus, not only did PUPs reuse the same infrastructure over long periods of time, but they frequently and consistently communicated with that infrastructure. This behavior is very likely associated with the types of abuse commonly associated with PUPs. For example, two common types of PUPs are adware and spyware, and both rely on some combination of surreptitiously gathering user information, monitoring user behavior, or displaying advertising. Each of these activities would likely require frequent communication with some common infrastructure—resulting in the usage patterns we observed.

Given the rapid growth in PUPs seen in our study, effectively addressing security concerns that stem from such software seems like a valuable future research direction. As discussed earlier, blocking PUPs at the network level poses challenges due to their use of benign, shared infrastructure. Thus, it might make sense to provide safeguards at the system level, but this too has challenges. Many PUPs are bundled alongside other legitimate applications, and this could be one of the reasons we observed a greater degree of binary polymorphism for such applications in our study—as discussed in Section 5.4. Furthermore, PUPs may have varying levels of integration with the software they're bundled with, and it may not always be the case that bundled PUPs can be safely removed without breaking the application they were installed with. The combination of binary polymorphism, bundled installers, and deep integration with bundled applications further add to the challenge of mitigating the spread of PUPs.

Clearly, PUPs pose difficult challenges for remediation, and their rise indicates that

current safeguards are not sufficient. A clear future research direction is the development of better methods, from both the network and system perspective, for dealing with this emerging threat.

6.3.2 Malware Sectors

Malware poses risks to all sorts of organizations. However, not all organizations share the same threat model nor are the potential repercussions of a malware infection the same between different organizations. One component that may affect the potential impact of a malware infection is the sector an organization belongs to. For example, a malware infection in a government network may have more dire consequences than a similar infection in a different sector. Furthermore, the types of malware targeting such networks may exhibit very different behaviors than more generic malware. Given the different network characteristics observed in our study, an interesting line of future research would be to examine how those differences vary across sectors.

One such behavior, the lifetime of a malware sample, would be interesting to measure across sector boundaries. We observed malware samples with very different lifetimes and lookup patterns in our study—some very long lived with very regular resolutions. Studying the lifetime of samples that affect a particular sector could provide insights into the types of malware families that are likely to target that sector. For instance, sensitive government or corporate networks might frequently be targets of advanced persistent threats, with the goal of exfiltrating sensitive data. Such malware might have a longer lifetime with fewer resolutions to avoid detection by security systems deployed on the network.

Our study also observed that many samples appeared to be active on the network for weeks or months before they were discovered and reported by the security community. Studying the sectors these particular samples target might provide insights into why they take so long to be discovered by the security community. For example, one hypothesis might be that such malware is discovered in more sensitive sectors first. These sectors may be

more reticent to release those samples back to the community as doing so would inform the adversary it had been discovered. Alternatively, the samples could be active in networks of sectors with less capable security teams, and thus, the samples simply go undetected for longer periods of time. Ultimately, understanding *why* these samples take so long to be discovered could help researchers build better defenses.

Studying how malware affects different sectors could be extremely valuable to the security community, but performing such a study is not without its challenges. To perform such a study with DNS, at a minimum, would require access to passive DNS data below the recursive. This would provide the client information necessary to associate lookups with a particular network, which could then be linked to a specific sector. Even better might be access to the authority for a major top level domain—as this would provide visibility into all resolutions for a particular domain. In this instance, the client information would be the recursive making the request on behalf of the DNS stub resolver. Many large organizations run their own recursives, and therefore, this may be enough information to identify the network associated with a specific sector. If an organization uses a third party recursive, it may still be possible to identify the network of the stub resolver using the EDNS client subnet (ECS) extension [72].

Performing a per sector analysis of malware network communication is a natural extension to the study presented in this thesis and could help answer new questions raised by some of our findings. Additionally, such a study serves as an excellent example of the type of future research enabled by the work presented in this thesis.

CHAPTER VII

CONCLUSION

Ultimately, the goal of each of study in this thesis has been to shed light onto different threats facing the Internet through empirical analysis. By performing large scale, longitudinal studies of these Internet threats; we provide insights that can aid in the creation of better defenses and guide future research directions. To achieve these insights, we performed three different large scale, longitudinal studies of Internet threats.

The first study presented addressed the emerging threat from mobile malware by empirically measuring malware communication in traffic from a large cellular ISP in the US. With the rise of smartphones and tablets, malware targeting these platforms has grown over time. However, prior to our study, it was not well understood how widely the mobile ecosystem was actually infected with mobile malware. Thus, a key finding of this study was that infection rates in the mobile ecosystem were much lower than previously believed and discovering that the network infrastructure was actually the same as that used by traditional malware.

The next study empirically analyzed the growing threat caused by expiring domain names. Domain names are often used as anchors of trust, where simple ownership of a domain is enough to attest one's identity. Unfortunately, domain ownership can change and, when this happens, the trust in that domain is implicitly inherited by the future owner—creating an opportunity for the new owner to abuse that trust. Existing security protections, such as DNS based blacklists and reputation systems, are impacted by this phenomenon as malicious actors can leverage the good reputation of an expired domain when they re-register the domain and repurpose it for abuse. Thus, a key result from this study was empirically measuring the growth of residual trust abuse by malware—showing consistent year over

year growth by malware.

Our final study presented a longitudinal analysis of nearly thirty million malware samples over a half decade. Given the wealth of malware collected over this period, a major goal of this study was to shed light on how the infrastructure and methods used by Internet miscreants have evolved over time. This enables us to confirm existing results, unearth new behaviors, and identify trends that may require further investigation. A key result from this study was the discovery that malware communication is frequently observed on the Internet weeks or months before corresponding malware samples are discovered—indicating defenses that rely on malware samples are unsuitable for early warning systems and may leave organizations vulnerable to new threats for extended periods of time.

In summary, this thesis provides three different empirical studies that provide valuable insights into different Internet threats. Chapter 3 presents a study of mobile malware that discusses the extent to which the mobile ecosystem is actually infected with malware—presenting a result which challenged the conventional wisdom at the time. In Chapter 4, our study of domain expirations demonstrated that the residual trust in domains is increasingly being abused by malware and is also the root cause for many seemingly disparate security problems. Finally, in Chapter 5, we presented our empirical study of 27M malware samples over a half decade, which demonstrated that malware samples are discovered weeks to months after we observe malware related network communication associated with a particular sample.

7.1 Considerations and Limitations

As we are measuring Internet threats, it is impossible to have perfect visibility into every aspect of a threat, and sometimes there are limitations to the data available for analysis. While these limitations do not invalidate the results of the study, it is important that they are known. In this section, we will discuss limitations encountered for each of the studies presented in this thesis.

7.1.1 Limitations of Mobile Malware Study

The goal of this study was to measure the extent to which the mobile ecosystem is actually infected with malware. This required access to a comprehensive set of mobile malware data and some means of estimating infected device populations.

7.1.1.1 Malware Datasets

The efficacy of any study on malware is limited to the quantity of *known* malware that the study can analyze. As seen in Table 3, we obtained mobile malware from the three most popular repositories at the time. While the total number of samples may seem small, these sources represented a comprehensive collection of known malware samples at the time of our study.

7.1.1.2 Estimating Infected Populations

One means of estimating device populations would be to collect information from security scanners on mobile devices. For the purposes of our study, obtaining such data from millions of real world devices would have been impossible. Not only would this have required users to install some sort of application on their mobile device, but that application would need sufficient privileges to scan the mobile device for infection. In essence, we would need access to data from a mobile antivirus vendor that was widely deployed across a large number of mobile devices. Unfortunately, mobile antivirus was still in its infancy and some studies suggested that it varied wildly in its detection efficacy [252]. Access to such scans would be potentially biased since users most apt to install antivirus applications may be more security conscious, and the results would also be biased on the detection efficacy of the particular AV vendor.

Instead, we relied on network indicators of compromise to identify mobile devices reaching out to infrastructure associated with mobile malware. As discussed in Section 3.3.3, we built a mobile blacklist sourced from three different sources: a security vendor, public

mobile malware reports, and domains extracted from malware samples. It is unlikely that this list contained *all* IOCs for every mobile malware sample, but given the diversity of sources upon which it was built, we are confident that this list was representative of the known mobile malware threats at the time.

7.1.1.3 Diversity of Network Data

Lastly, we used passive DNS data collected from a major US cellular ISP for our network analysis. As a result, our study is inherently biased towards an area with strong first party mobile markets (i.e., Google Play, iOS App Store). Given the timing of this study, the US market was currently one of largest markets for smartphone adoption, and we are confident in these results for that market. An interesting direction for future research could be a similar study in other countries that lack access to strong first party mobile markets.

7.1.2 Limitations of Expired Domain Study

The goal of this study was to understand the threat from expired domains by empirically measuring their abuse. To perform this study, we needed to the ability to identify domain expirations, observe when expired domains were queried by malware, and observe when malware domains were first resolved.

7.1.2.1 WHOIS Data

To identify changes in domain ownership, WHOIS data is a natural choice since it provides a registry of who owns a particular domain at any point in time. Unfortunately, WHOIS data has a number of problems. First, the data in WHOIS is not verified, which means domain owners can put anything they want in their WHOIS records. Second, WHOIS data is frequently privacy protected, containing only information about the service providing the privacy service. Third, WHOIS data is difficult to obtain at scale because each registrar imposes its own policies on automated querying of WHOIS data. Exceeding these restrictions will result in the offending IP address being blacklisted for some variable period of

time. Therefore, in our study, we were only able to make limited use of WHOIS in verifying results—limiting our ability to build a true classification system for domain ownership changes.

7.1.3 Limitations of Longitudinal Malware Study

The goal of this study was to shed light on how the infrastructure and methods used by malware have evolved over time. This study required access to a large set of malware samples collected over an extended period of time. Consequently, most of the limitations of this study surround the data necessary to perform this study.

7.1.3.1 Malware Datasets

As detailed in Section 5.2, our study used network signal collected from dynamic malware analysis of nearly 27M samples over a half decade, and these executions results were sourced from three different malware feeds. Dynamic analysis of malware has a number of known limitations. For example, dynamic execution environments run for some predefined quantum of time before exiting—a behavior which is necessary due to the halting problem. If this quantum is too short, the execution of the malware may be exited before any sort of malicious behavior has been triggered. This is an inherent limitation of dynamic malware analysis. In our study, we are not concerned with measuring the limitations of dynamic execution environments; instead, we simply want the network signal that *is* collected from such analysis. While the details of each feed may differ, to the best of our knowledge they all are adhering to state of the art practices for dynamic malware analysis.

7.1.3.2 Passive DNS Data

We complement our analysis with passive DNS data from a major ISP in the US. This data allows us to observe when we first see resolutions to malware domains in a real world network, as discussed in Section 5.5.2.1. Since we only have access to DNS resolutions, we cannot inspect the communication payloads being sent to the infrastructure a malware

domain resolves to. In our analysis, we do try and exclude resolutions that are unlikely to be associated with malware communication by excluding communication before a domain expired—which are likely instances of malware abusing positive residual trust abuse as discussed in detail in Chapter 4. We also analyze domain lifetimes in Section 5.5.2.2 to understand the resolution behavior of domains that are resolved weeks or months before their corresponding malware is discovered, and we find that most domains are looked up frequently over their lifetime—suggesting periodic lookup behavior consistent with malware activity. Still, we cannot be absolutely certain why a domain is resolved prior to discovering a malware sample since we cannot analyze the communication that followed a given DNS resolution.

7.2 Closing Remarks

This thesis studied several emerging and existing threats through empirical analysis with passive DNS data. These studies resulted in three key insights that were previously unknown to the security community. First, the study of mobile malware empirically demonstrated that very few mobile devices appear to be infected with mobile malware—challenging the conventional wisdom of the time—and that mobile threats look similar from the perspective of network infrastructure. The second study in this thesis analyzed how residual trust in domains is abused by malicious actors. This study not only showed that residual trust abuse is the underlying cause of many seemingly disparate security problems, but it is also a threat that is increasingly being leveraged by malware—showing consistent growth year over year. Lastly, our longitudinal study of nearly 27M malware samples over half a decade demonstrated that malware samples are frequently discovered weeks or months after the threat is visible on the network—suggesting that systems that rely on malware may result in large windows of vulnerability for organizations that rely on them. Not only do the studies in this unearth valuable insights into several existing and emerging Internet threats, but they underscore how DNS provides a platform agnostic data source for studying Internet threats.

In summary, the insights on Internet threats presented in this thesis should guide future research and provide valuable insights that help build better defenses.

REFERENCES

[1] "DNSDynamic - Absolutely Free Dynamic DNS." https://www.dnsdynamic.org, 2016.

[2] "Heytell - instant voice messaging." http://heytell.com/front.html.

[3] "Internet corporations for assigned names and numbers." http://www.icann.org.

[4] ".mobi tld sponsorship agreement." http://www.icann.org/tlds/agreements/mobi/.

[5] "Public suffix list." http://publicsuffix.org.

[6] "New stld rfp application (.mobi)." http://archive.icann.org/en/tlds/stld-apps-19mar04/mobi.htm, 2004.

[7] "Android/DroidKungFu.A!tr." http://www.fortiguard.com/encyclopedia/virus/android_droidkungfu.a!tr.html, June 2011.

[8] "Android/FakeDoc.A!tr." http://www.fortiguard.com/av/VID3304615, Dec. 2011.

[9] "Android/Steek.A!tr." http://www.fortiguard.com/av/VID3458224, Jan. 2012.

[10] "APT1: Exposing One of China's Cyber Espionage Units," tech. rep., Mandiant, 2013. http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf.

[11] "New Spambot In Town Using Compromised Websites To Send Spam," August 2013. https://www.abuse.ch/?tag=rodecap.

[12] "Presidential Policy Directive – Critical Infrastructure Security and Resilience." http://www.whitehouse.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil, February 2013.

[13] "Millions of dynamic DNS users suffer after Microsoft seizes No-IP domains." http://arstechnica.com/security/2014/06/millions-of-dymanic-dns-users-suffer-after-microsoft-seizes-no-ip-domains, 2014.

[14] "Putter Panda: PLA Army 3rd Department 12th Bureau Unit 61486," tech. rep., CrowdStrike, Inc., 2014. http://resources.crowdstrike.com/putterpanda/.

[15] "Backdoor:Win32/Polif.A." http://www.microsoft.com/security/portal/threat/encyclopedia/Entry.aspx?Name=Backdoor%3AWin32%2FPolif.A#tab=2, 2015.

[16] "Data breach investigations report," tech. rep., Verizon, 2015.

[17] "Detailed domain name information and archives in one place." http://www.domainhistory.net/, 2015.

[18] "Domain Blacklist: abuse.ch." http://www.abuse.ch/, 2015.

[19] "Domain Blacklist: Blackhole DNS." http://www.malwaredomains.com/wordpress/?page_id=6, 2015.

[20] "Domain Blacklist: driveby." http://www.blade-defender.org/eval-lab/, 2015.

[21] "Domain Blacklist: hphosts." http://hosts-file.net/?s=Download, 2015.

[22] "Domain Blacklist: itmate." http://vurl.mysteryfcm.co.uk/, 2015.

[23] "Domain Blacklist: sagadc." http://dns-bh.sagadc.org/, 2015.

[24] "Domain Blacklist: SANS." https://isc.sans.edu/suspicious_domains.html, 2015.

[25] "Domain Graveyard." http://domaingraveyard.com/, 2015.

[26] "Malware Domain List." http://www.malwaredomainlist.com/forums/index.php?topic=3270.0, 2015.

[27] "TrojanDownloader:Win32/Nivdort.C." http://www.microsoft.com/security/portal/threat/encyclopedia/entry.aspx?Name=TrojanDownloader:Win32/Nivdort.C#tab=2, 2015.

[28] "Whoi.is." https://who.is/domain-history/, 2015.

[29] "Whois History." https://www.domaintools.com/research/whois-history/, 2015.

[30] "AdMob." http://www.admob.com, 2016.

[31] "DGArchive." https://dgarchive.caad.fkie.fraunhofer.de/, 2016.

[32] "DNS-BH - Malware Domain Blocklist." http://www.malwaredomains.com/?cat=140, 2016.

[33] "DynDNS." http://dyn.com/remote-access, 2016.

[34] "Find Domain Names for Your DynDNS Pro Plan." http://dyn.com/remote-access/domain-names, 2016.

[35] "FreeDNS." https://freedns.afraid.org, 2016.

[36] "No-ip - free dynamic dns." http://www.noip.com, 2016.

[37] "VirusTotal - Free Online Virus, Malware, and URL Scanner." http://www.virustotal.com, 2016.

[38] "Android security 2017 year in review," tech. rep., Google, 2018.

[39] AGTEN, P., JOOSEN, W., PIESSENS, F., and NIKIFORAKIS, N., "Seven months' worth of mistakes: A longitudinal study of typosquatting abuse," in *Proceedings of the 22nd Network and Distributed System Security Symposium (NDSS)*, 2015.

[40] ALEXA, "The web information company." http://www.alexa.com/, 2007.

[41] ALEXA INTERNET, I., "Alexa the web information company," 2011.

[42] AMRUTKAR, C., TRAYNOR, P., and VAN OORSCHOT, P. C., "Measuring ssl indicators on mobile browsers: Extended life, or end of the road?," in *International Conference on Information Security*, pp. 86–103, Springer, 2012.

[43] ANTONAKAKIS, M., PERDISCI, R., DAGON, D., LEE, W., and FEAMSTER, N., "Building a Dynamic Reputation System for DNS," in *Proceedings of the 19th USENIX Security Symposium (SECURITY)*, 2010.

[44] ANTONAKAKIS, M., PERDISCI, R., LEE, W., VASILOGLOU, N., and DAGON, D., "Detecting Malware Domains at the Upper DNS Hierarchy," in *Proceedings of the 20th USENIX Security Symposium (SECURITY)*, 2011.

[45] ANTONAKAKIS, M., APRIL, T., BAILEY, M., BERNHARD, M., BURSZTEIN, E., COCHRAN, J., DURUMERIC, Z., HALDERMAN, J. A., INVERNIZZI, L., KALLITSIS, M., KUMAR, D., LEVER, C., MA, Z., MASON, J., MENSCHER, D., SEAMAN, C., SULLIVAN, N., THOMAS, K., and ZHOU, Y., "Understanding the mirai botnet," in *Proceedings of the 26th USENIX Security Symposium (SECURITY)*, 2017.

[46] ANTONAKAKIS, M., DEMAR, J., STEVENS, K., and DAGON, D., "Unveiling the Network Criminal Infrastructure of TDSS/TDL4 DGAv14: A case study on a new TDSS/TDL4 variant.," tech. rep., Damballa Inc.,Georgia Institute of Technology (GTISC), 2012.

[47] ANTONAKAKIS, M., ELISAN, C., DE MATA, A., OLLMANN, G., and WU, E., "The IMDDOS Botnet: Discovery and Analysis," tech. rep., Damballa Inc.,Georgia Institute of Technology (GTISC), 2010.

[48] ANTONAKAKIS, M., PERDISCI, R., DAGON, D., LEE, W., and FEAMSTER, N., "Building a Dynamic Reputation System for DNS," in *Proceedings of the 19th USENIX Conference on Security (USENIX Security)*, August 2010.

[49] ANTONAKAKIS, M., PERDISCI, R., LEE, W., VASILOGLOU, N., and DAGON, D., "Detecting Malware Domains in the Upper DNS Hierarchy," in *Proceedings of the 20th USENIX Conference on Security (USENIX Security)*, August 2011.

[50] ANTONAKAKIS, M., PERDISCI, R., NADJI, Y., VASILOGLOU, N., ABU-NIMEH, S., LEE, W., and DAGON, D., "From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware," in *Proceedings of the 21st USENIX Security Symposium (SECURITY)*, 2012.

[51] ANTONAKAKIS, M., PERDISCI, R., NADJI, Y., VASILOGLOU II, N., ABU-NIMEH, S., LEE, W., and DAGON, D., "From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware," in *Proceedings of the 21st USENIX Conference on Security (USENIX Security)*, August 2012.

[52] ANTONAKAKIS, M., STONE-GROSS, B., DEMAR, J., STEVENS, K., and DAGON, D., "Unveiling The Latest Variant of Pushdo Mv20: A case study on the new Pushdo-DGA.," tech. rep., Damballa Inc., Dell SecureWorks CTU, Georgia Institute of Technology (GTISC), 2012.

[53] APNIC, "Reverse DNS Delegation." `https://www.apnic.net/manage-ip/manage-resources/reverse-dns`, 2009.

[54] ARENDS, R., AUSTEIN, R., LARSON, M., MASSEY, D., and ROSE, S., "DNS security introduction and requirements," tech. rep., RFC 4033, March, 2005.

[55] BAILEY, M., OBERHEIDE, J., ANDERSEN, J., MAO, Z. M., JAHANIAN, F., and NAZARIO, J., "Automated Classification and Analysis of Internet Malware," in *Proceedings of the 10th International Symposium on Recent Advances in Intrusion Detection (RAID)*, 2007.

[56] BANDHAKAVI, S., KING, S. T., MADHUSUDAN, P., and WINSLETT, M., "VEX: Vetting Browser Extensions For Security Vulnerabilities," in *Proceedings of the 19th USENIX Conference on Security (USENIX Security)*, August 2010.

[57] BARTH, A., FELT, A. P., SAXENA, P., and BOODMAN, A., "Protecting Browsers from Extension Vulnerabilities," in *Proceedings of the 17th Network and Distributed System Security Symposium (NDSS)*, February 2010.

[58] BAYER, U., HABIBI, I., BALZAROTTI, D., KIRDA, E., and KRUEGEL, C., "A View on Current Malware Behaviors," in *Proceedings of the 2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2009.

[59] BBC NEWS, "Malicious app penetrates iTunes store to test security." http://www.bbc.co.uk/news/technology-15635408, 2012.

[60] BELLOVIN, S. M., "Problem Areas for the IP Security Protocols," in *Proceedings of the 6th USENIX Conference on Security (USENIX Security)*, July 1996.

[61] BILGE, L., KIRDER, E., KRUEGEL, C., and BALDUZZI, M., "Exposure: Finding malicious domains using passive DNS analysis," in *Proceedings of the 18th Network and Distributed System Security Symposium (NDSS)*, 2011.

[62] BILGE, L., KIRDA, E., KRUEGEL, C., and BALDUZZI, M., "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis.," in *Proceedings of the 18th Network and Distributed System Security Symposium (NDSS)*, February 2011.

[63] BROIDO, A., NEMETH, E., and KC CLAFFY, "Spectroscopy of DNS Update Traffic." https://www.caida.org/publications/papers/2003/dnsspectroscopy/dnsspectroscopy.pdf, 2005.

[64] CABALLERO, J., JOHNSON, N. M., MCCAMANT, S., and SONG, D., "Binary Code Extraction and Interface Identification for Security Applications," in *Proceedings of the 14th Network and Distributed System Security Symposium (NDSS)*, 2010.

[65] CANALI, D., BALZAROTTI, D., and FRANCILLON, A., "The Role of Web Hosting Providers in Detecting Compromised Websites," in *Proceedings of the 22nd International World Wide Web Conference (WWW)*, (Rio de Janeiro, Brazil), May 2013.

[66] CANALI, D., COVA, M., VIGNA, G., and KRUEGEL, C., "Prophiler: A Fast Filter for the Large-scale Detection of Malicious Web Pages," in *Proceedings of the 20th International Conference on World Wide Web (WWW)*, March 2011.

[67] CARLINI, N., FELT, A. P., and WAGNER, D., "An Evaluation of the Google Chrome Extension Security Architecture," in *Proceedings of the 21st USENIX Conference on Security (USENIX Security)*, August 2012.

[68] CATHAL MULLANEY and JEET MORPARIA, "Android.Tonclank." http://www.symantec.com/security_response/writeup.jsp?docid=2011-061012-4545-99, June 2011.

[69] CHAKRADEO, S., REAVES, B., TRAYNOR, P., and ENCK, W., "Mast," *Proceedings of the 6th ACM conference on Security and Privacy in Wireless and Mobile Networks (WISEC)*, 2013.

[70] CHIN, E., FELT, A., GREENWOOD, K., and WAGNER, D., "Analyzing Inter-Application Communication in Android," in *Proceedings of International Conference on Mobile Systems, Applications, and Services*, 2011.

[71] CONSORTIUM, I. S., "Sie@isc: Security information exchange." https://sie.isc.org, 2004.

[72] CONTAVALLI, C., VAN DER GAAST, W., LAWRENCE, D., and KUMARI, W., "Client subnet in dns queries," RFC 7871, RFC Editor, May 2016.

[73] COOKE, E., JAHANIAN, F., and MCPHERSON, D., "The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets," in *Proceedings of Steps to Reducing Unwanted Traffic on the Internet Workshop*, 2005.

[74] DAGON, D., ANTONAKAKIS, M., VIXIE, P., JINMEI, T., and LEE, W., "Increased DNS Forgery Resistance Through 0x20-bit Encoding: SecURItY viA LeET QueRieS," in *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS)*, 2008.

[75] DAGON, D., MARTIN, T., and STARNER, T., "Mobile phones as computing devices: The viruses are coming!," *IEEE Pervasive Computing*, vol. 3, pp. 11–15, Oct. 2004.

[76] DAGON, D., ANTONAKAKIS, M., VIXIE, P., JINMEI, T., and LEE, W., "Increased DNS Forgery Resistance Through 0x20-bit Encoding: Security via Leet Queries," in *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS)*, October 2008.

[77] DAIGLE, L., "WHOIS Protocol Specification." RFC 3912 (Draft Standard), Sept. 2004.

[78] DINABURG, A., "Bitsquatting: Dns hijacking with exploitation," tech. rep., Raytheon Company, 2011.

[79] DNSBL, S., "Fighting spam by finding and listing exploitable servers," 2007.

[80] ENCK, W., GILBERT, P., CHUN, B., COX, L., JUNG, J., MCDANIEL, P., and SHETH, A., "Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones," in *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation*, 2010.

[81] ENCK, W., OCTEAU, D., MCDANIEL, P., and CHAUDHURI, S., "A study of Android application security," in *Proceedings of the 20th USENIX Security Symposium*, 2011.

[82] ENCK, W., ONGTANG, M., and MCDANIEL, P., "Understanding Android security," *IEEE Security and Privacy Magazine*, vol. 7, no. 1, pp. 50–57, 2009.

[83] ENCK, W., ONGTANG, M., and MCDANIEL, P., "On Lightweight Mobile Phone Application Certification," in *Proceedings of the ACM Conference on Computer and Communications Security*, 2009.

[84] ERICKSON, D., CASADO, M., and MCKEOWN, N., "The Effectiveness of Whitelisting: a User-Study," in *Proceedings of the 5th Conference on Email and Anti-Spam (CEAS)*, August 2008.

[85] ERMAN, J., GERBER, A., RAMADRISHNAN, K. K., SEN, S., and SPATSCHECK, O., "Over the top video: the gorilla in cellular networks," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, (New York, NY, USA), pp. 127–136, ACM, 2011.

[86] ESCROW.COM, I., "Pay online securely with escrow.com." https://www.escrow.com/, 2017.

[87] ESET, "Eset threat trends for 2013: Growth of mobile malware; botnets; cloud and leaks." https://www.eset.com/us/about/announcements/eset-threat-trends-for-2013-growth-of-mobile-malware-botnets-cloud-and-leaks/, December 2012.

[88] FALAKI, H., LYMBEROPOULOS, D., MAHAJAN, R., KANDULA, S., and ESTRIN, D., "A first look at traffic on smartphones," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, IMC '10, (New York, NY, USA), pp. 281–287, ACM, 2010.

[89] FELEGYHAZI, M., KREIBICH, C., and PAXSON, V., "On the Potential of Proactive Domain Blacklisting," in *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More (LEET)*, April 2010.

[90] FELT, A., CHIN, E., HANNA, S., SONG, D., and WAGNER, D., "Android permissions demystified," in *Proceedings of the ACM Conference on Computer and Communication Security*, 2011.

[91] FELT, A., FINIFTER, M., CHIN, E., HANNA, S., and WAGNER, D., "A survey of mobile malware in the wild," in *ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, 2011.

[92] FELT, A., GREENWOOD, K., and WAGNER, D., "The effectiveness of application permissions," in *Proceedings of the USENIX Conference on Web Application Development*, 2011.

[93] FELT, A., WANG, H., MOSCHUK, A., HANNA, S., and CHIN, E., "Permission re-delegation: Attacks and defenses," in *Proceedings of the 20th USENIX Security Symposium*, 2011.

[94] FELT, A. P. and WAGNER, D., "Phishing on mobile devices," in *Proceedings of the Web 2.0 Security and Privacy 2011 workshop (W2SP 2011)*, 2011.

[95] FORESAN, C., "Proof-of-concept app exploiting ios security flaw gets researcher in trouble with apple," 11 2011.

[96] FORTINET, "Android/sndapp.a." http://www.fortiguard.com/av/VID3148366, 2011.

[97] FOUNDATION, T. A. S., "The Apache Cassandra Project." http://cassandra.apache.org, 2011.

[98] FRIED, I., "Droid dream malware latest sign Android attacks are on the rise," 3 2011.

[99] FS-ISAC, "Financial Services Information Sharing and Analysis Center." https://www.fsisac.com/, 2015.

[100] GDPR, "Home page of eu gdpr." https://www.eugdpr.org/, 2017.

[101] GEMBER, A., ANAND, A., and AKELLA, A., "A comparative study of handheld and non-handheld traffic in campus Wi-Fi networks," in *Proceedings of the 12th international conference on Passive and active measurement*, PAM'11, (Berlin, Heidelberg), pp. 173–183, Springer-Verlag, 2011.

[102] GOOGLE, "Android market," 2011.

[103] GOOGLE, "Google play." https://play.google.com/store, 2017.

[104] GOOGLE, "Linking to google play." https://developer.android.com/distribute/marketing-tools/linking-to-google-play.html, 2017.

[105] GRIER, C., BALLARD, L., CABALLERO, J., CHACHRA, N., DIETRICH, C. J., LEVCHENKO, K., MAVROMMATIS, P., MCCOY, D., NAPPA, A., PITSILLIDIS, A., PROVOS, N., RAFIQUE, M. Z., RAJAB, M. A., ROSSOW, C., THOMAS, K., PAXSON, V., SAVAGE, S., and VOELKER, G. M., "Manufacturing Compromise: The Emergence of Exploit-as-a-Service," in *Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS)*, 2012.

[106] GROSS, B., COVA, M., CAVALLARO, L., GILBERT, B., SZYDLOWSKI, M., KEMMERER, R., KRUEGEL, C., and VIGNA, G., "Your botnet is my botnet: Analysis of a botnet takeover," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, 2009.

[107] GRYBOSKI, M., "Facebook Clarifies Reason for Blocking Kirk Cameron's "Unstoppable"," July 2013.

[108] GU, G., ZHANG, J., and LEE, W., "Botsniffer: Detecting botnet command and control channels in network traffic," in *Proceedings of the 15th Network and Distributed System Security Symposium*, 2008.

[109] GUNDERT, L., "Dynamic Detection of Malicious DDNS." http://blogs.cisco.com/security/dynamic-detection-of-malicious-ddns, 2014.

[110] GUO, C., WANG, H. J., and ZHU, W., "Smart Phone Attacks and Defenses," in *Proceedings of Third ACM Workshop on Hot Topics in Networks (HotNets-III)*, 2004.

[111] HAN, X., KHEIR, N., and BALZAROTTI, D., "The Role of Cloud Services in Malicious Software: Trends and Insights," in *Proceedings of the 12th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, July 2015.

[112] HAO, S., FEAMSTER, N., and PANDRANGI, R., "An Internet Wide View into DNS Lookup Patterns," tech. rep., Verisign Labs, 2010.

[113] HAO, S., KANTCHELIAN, A., MILLER, B., PAXSON, V., and FEAMSTER, N., "PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration," in *Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS)*, 2016.

[114] HAO, S., THOMAS, M., PAXSON, V., FEAMSTER, N., KREIBICH, C., GRIER, C., and HOLLENBECK, S., "Understanding the Domain Registration Behavior of Spammers," in *Proceedings of the 2013 Internet Measurement Conference (IMC)*, 2013.

[115] HAO, S., THOMAS, M., PAXSON, V., FEAMSTER, N., KREIBICH, C., GRIER, C., and HOLLENBECK, S., "Understanding the Domain Registration Behavior of Spammers," in *Proceedings of the 2013 Conference on Internet Measurement Conference (IMC)*, October 2013.

[116] HOLLENBECK, S., "Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP)." RFC 3915 (Proposed Standard), Sept. 2004.

[117] HOLLENBECK, S., "Extensible Provisioning Protocol (EPP)." RFC 5730 (INTERNET STANDARD), Aug. 2009.

[118] HOLZ, T., GORECKI, C., RIECK, K., and FREILING, F., "Measuring and detecting fast-flux service networks," in *Proceedings of 16th Network and Distributed Systems Security Symposium (NDSS)*, 2008.

[119] HORN, L., "Report: Android market reaches 500,000 apps," 10 2011.

[120] HOUSLEY, R., CURRAN, J., HUSTON, G., and CONRAD, D., "The Internet Numbers Registry System." RFC 7020 (Informational), Aug. 2013.

[121] HOVER, "How to calculate your domain name's value." `https://www.hover.com/blog/find-out-domain-name-value/`, 2016.

[122] HUANG, W. and STOKES, J. W., "MtNet: A Multi-Task Neural Network for Dynamic Malware Classification," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, 2016.

[123] ICANN, "Discussion Paper: Redemption Grace Periods for Deleted Names." `https://archive.icann.org/en/registrars/redemption-proposal-14feb02.htm`, February 2002.

[124] ICANN, "EPP Status Codes." https://www.icann.org/resources/pages/epp-status-codes-2014-06-16-en, 2015.

[125] ICANN, "Expired Registration Recovery Policy." https://www.icann.org/resources/pages/errp-2013-02-28-en, 2015.

[126] ICANN, "Uniform Domain-Name Dispute-Resolution Policy." https://www.icann.org/resources/pages/help/dndr/udrp-en, 2015.

[127] ICANN, "Data protection/privacy issues." https://www.icann.org/dataprotectionprivacy, 2017.

[128] ICANN, "Uniform domain-name dispute-resolution policy." https://www.icann.org/resources/pages/help/dndr/udrp-en, 2017.

[129] IEDEMSKA, J., STRINGHINI, G., KEMMERER, R., KRUEGEL, C., and VIGNA, G., "The Tricks of the Trade: What Makes Spam Campaigns Successful?," in *IEEE CS Security and Privacy Workshops (SPW)*, May 2014.

[130] INC., A., "Apple's App Store Downloads Top Three Billion," 1 2010.

[131] INC., A., "The App Store - There's an app for that. Over 500,000, actually.," 2011.

[132] INC., A., "Creating easy-to-read short links to the app store for your apps and company." https://developer.apple.com/library/content/qa/qa1633/_index.html, 2013.

[133] INC., A., "App store." https://www.apple.com/ios/app-store/, 2017.

[134] INVERNIZZI, L., MISKOVIC, S., TORRES, R., KRUEGEL, C., SAHA, S., VIGNA, G., LEE, S.-J., and MELLIA, M., "Nazca: Detecting Malware Distribution in Large-Scale Networks.," in *Proceedings of the 18th Network and Distributed Systems Symposium (NDSS)*, 2014.

[135] ISAAC, M., "Android market hits 10 billion downloads, kicks off app sale," 12 2011.

[136] JUNG, J., SIT, E., BALAKRISHNAN, H., and MORRIS, R., "DNS Performance and the Effectiveness of Caching," *IEEE/ACM Transactions on Networking*, 2002.

[137] KAPRAVELOS, A., GRIER, C., CHACHRA, N., KRUEGEL, C., VIGNA, G., and PAXSON, V., "Hulk: Eliciting Malicious Behavior in Browser Extensions," in *Proceedings of the 23rd USENIX Conference on Security (USENIX Security)*, Aug. 2014.

[138] KASSNER, M., "Google Play: Android's Bouncer can be pwned." http://www.techrepublic.com/blog/security/google-play-androids-bouncer-can-be-pwned/8053, 2012.

[139] KINTIS, P., MIRAMIRKHANI, N., LEVER, C., CHEN, Y., ROMERO-GÓMEZ, R., PITROPAKIS, N., NIKIFORAKIS, N., and ANTONAKAKIS, M., "Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 08 2017.

[140] KOLBITSCH, C., HOLZ, T., KRUEGEL, C., and KIRDA, E., "Inspector Gadget: Automated Extraction of Proprietary Gadgets from Malware Binaries," in *Proceedings of the 31st IEEE Symposium on Security and Privacy (OAKLAND)*, 2010.

[141] KOTZIAS, P., BILGE, L., and CABALLERO, J., "Measuring PUP Prevalence and PUP Distribution through Pay-Per-Install Services," in *Proceedings of the 25th USENIX Security Symposium (SECURITY)*, 2016.

[142] KOTZIAS, P., MATIC, S., RIVERA, R., and CABALLERO, J., "Certified PUP: Abuse in Authenticode Code Signing," in *Proceedings of the 22nd ACM Conference on Computer and Communication Security (CCS)*, 2015.

[143] KÜHRER, M., ROSSOW, C., and HOLZ, T., "Paint It Black: Evaluating the Effectiveness of Malware Blacklists," in *Proceedings of the 18th International Syposium on Research in Attacks, Intrusions, and Defenses (RAID)*, 2014.

[144] KWON, B. J., MONDAL, J., JANG, J., BILGE, L., and DUMITRAS, T., "The Dropper Effect: Insights into Malware Distribution with Downloader Graph Analytics," in *Proceedings of 22nd ACM Conference on Computer and Communications Security (CCS)*, 2015.

[145] KWON, B. J., SRINIVAS, V., DESHPANDE, A., and DUMITRAS, T., "Catching Worms, Trojan Horses and PUPs: Unsupervised Detection of Silent Delivery Campaigns," in *Proceedings of the 20th Network and Distributed Systems Security Symposium (NDSS)*, 2017.

[146] LAB, I. S. S., "Anubis - Malware Analysis for Unknown Binaries." `http://anubis.iseclab.org`, 2016.

[147] LEE, S.-W., PARK, J.-S., LEE, H.-S., and KIM, M.-S., "A study on smart-phone traffic analysis," in *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, pp. 1 –7, Sept. 2011.

[148] LEE, S.-W., PARK, J.-S., LEE, H.-S., and KIM, M.-S., "A study on smart-phone traffic analysis," in *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, pp. 1 –7, Sept. 2011.

[149] LEE, S. and KIM, J., "WarningBird: Detecting Suspicious URLs in Twitter Stream," in *Proceedings of the 9th Network and Distributed System Security Symposium (NDSS)*, February 2012.

[150] LEVER, C., ANTONAKAKIS, M., REAVES, B., TRAYNOR, P., and LEE, W., "Core of the matter: Analyzing malicious traffic in cellular carriers," in *Proceedings of the 20th Network and Distributed System Security Symposium (NDSS)*, 2013.

[151] LEVER, C., KOTZIAS, P., BALZAROTTI, D., CABALLERO, J., and ANTONAKAKIS, M., "A lustrum of malware network communication: Evolution and insights," in *Proceedings of the 38th IEEE Symposium on Security and Privacy (OAKLAND)*, 2017.

[152] LEVER, C., WALLS, R., NADJI, Y., DAGON, D., and ANTONAKAKIS, M., "Domain-z: 28 registrations later," in *Proceedings of the 37th IEEE Symposium on Security and Privacy (OAKLAND)*, 2016.

[153] LEVER, C., WALLS, R., NADJI, Y., DAGON, D., MCDANIEL, P., and ANTONAKAKIS, M., "Domain-Z: 28 Registrations Later," in *Procededings of the 37th IEEE Symposium on Security and Privacy (OAKLAND)*, 2016.

[154] LINDORFER, M., NEUGSCHWANDTNER, M., WEICHSELBAUM, L., FRATANTONIO, Y., VAN DER VEEN, V., and PLATZER, C., "Andrubis - 1,000,000 Apps Later: A View on Current Android Malware Behaviors," in *Proceedings of the 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*, 2014.

[155] LIU, C. and ALBITZ, P., *DNS and BIND*. O'Reilly Media, 5th edition ed., 2006.

[156] LIU, S., FOSTER, I., SAVAGE, S., and VOELKER, G. M., "Who is .com? Learning to Parse WHOIS Records," in *Proceedings of the 2015 Conference on Internet Measurement Conference (IMC)*, October 2015.

[157] MA, J., SAUL, L., SAVAGE, S., and VOELKER, G., "Beyond blacklists: Learning to detect malicious web sites from suspicious urls," in *Proceedings of the SIGKDD Conference*, 2009.

[158] MA, J., SAUL, L. K., SAVAGE, S., and VOELKER, G. M., "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, June 2009.

[159] MARSLAND, S., *Machine Learning: An Algorithmic Perspective*. Chapman and Hall/CRC, 1st edition ed., 2009.

[160] METCALF, L. and SPRING, J. M., "Blacklist Ecosystem Analysis: Spanning Jan 2012 to Jun 2014," in *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security (WISCS)*, 2015.

[161] MICRO, T., "Update: Mobile threats on the rise." `https://blog.trendmicro.com/update-mobile-threats-on-the-rise/`, 2017.

[162] MICROSOFT, "Viewing Vobfus Infections from Above." https://trac.torproject.org/projects/tor/ticket/7349, 2013.

[163] MITCHELL, R. L., "Domain-name wars: Rise of the cybersquatters." https://www.cio.com/article/2424985/infrastructure/domain-name-wars--rise-of-the-cybersquatters.html, 2009.

[164] MOCKAPETRIS, P., "Domain names - concepts and facilities." RFC 1034 (Standard), Nov. 1987.

[165] MOCKAPETRIS, P., "Domain names - implementation and specification." RFC 1035 (Standard), Nov. 1987.

[166] MOHAISEN, A. and ALRAWI, O., "AV-Meter: An Evaluation of Antivirus Scans and Labels," in *Proceedings of the 11th Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2014.

[167] MOORE, T. and CLAYTON, R., "The Ghosts of Banking Past: Empirical Analysis of Closed Bank Websites," in *Financial Cryptography and Data Security*, March 2014.

[168] MOSCHUK, A., BRAGIN, T., GRIBBLE, S. D., and LEVY, H., "A Crawler-based Study of Spyware in the Web," in *Proceedings of the 10th Network and Distributed System Security Symposium (NDSS)*, 2006.

[169] MOSER, A., KRUEGEL, C., and KIRDA, E., "Exploring multiple execution paths for malware analysis," in *Proceedings of the 28th IEEE Symposium on Security and Privacy (OAKLAND)*, 2007.

[170] MOSKOWITZ, B., KARRENBERG, D., DE GROOT, G. J., and LEAR, E., "Address Allocation for Private Internets." https://tools.ietf.org/rfc/rfc1918.txt, 1996.

[171] NADJI, Y., ANTONAKAKIS, M., PERDISCI, R., DAGON, D., and LEE, W., "Beheading Hydras: Performing Effective Botnet Takedowns," in *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS)*, 2013.

[172] NADJI, Y., ANTONAKAKIS, M., PERDISCI, R., DAGON, D., and LEE, W., "Beheading Hydras: Performing Effective Botnet Takedowns," in *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS)*, November 2013.

[173] NADJI, Y., ANTONAKAKIS, M., PERDISCI, R., and LEE, W., "Understanding the Prevalence and Use of Alternative Plans in Malware with Network Games," in *Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC)*, 2011.

[174] NAPPA, A., RAFIQUE, M. Z., and CABALLERO, J., "Driving in the Cloud: An Analysis of Drive-By Download Operations and Abuse Reporting," in *Proceedings of the 10th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2013.

[175] NAPPA, A., XU, Z., CABALLERO, J., and GU, G., "CyberProbe: Towards Internet-Scale Active Detection of Malicious Servers," in *Proceedings of the 18th Network and Distributed System Security Symposium (NDSS)*, 2014.

[176] NINO FRED GUTIERREZ, "Android.Gonesixty." `http://www.symantec.com/security\_response/writeup.jsp?docid=2011-093001-2649-99`, Sept. 2011.

[177] OBERHAUS, D., "Whatis going to happen with whois?." `https://motherboard.vice.com/en_us/article/vbpgga/whois-gdpr-europe-icann-registrar`, 2018.

[178] OPENDNS, "Opendns — internet navigation and security," 2011.

[179] PELLEG, D. and MOORE, A. W., "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, (San Francisco, CA, USA), pp. 727–734, Morgan Kaufmann Publishers Inc., 2000.

[180] PERDISCI, R., CORONA, I., DAGON, D., and LEE, W., "Detecting malicious flux service networks through passive analysis of recursive DNS traces," in *Proceedings of the 25th Annual Compuer Security Applications Conference*, 2009.

[181] PIOTR KRYSIUK, "Android.Ggtracker." `http://www.symantec.com/security\_response/writeup.jsp?docid=2011-062208-5013-99`, June 2011.

[182] PLOHMANN, D., YAKDAN, K., KLATT, M., BADER, J., and GERHARDS-PADILLA, E., "A Comprehensive Measurement Study of Domain Generating Malware," in *Proceedings of the 25th USENIX Security Symposium (SECURITY)*, 2016.

[183] POLYCHRONAKIS, M., MAVROMMATIS, P., and PROVOS, N., "Ghost Turns Zombie: Exploring the Life Cycle of Web-Based Malware," in *Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[184] PORRAS, P., SAIDI, H., and YEGNESWARAN, V., "An Analysis of Conficker's Logic and Redezvous Points," tech. rep., SRI International, 2009.

[185] PRAKASH, P., KUMAR, M., KOMPELLA, R. R., and GUPTA, M., "Phishnet: Predictive Blacklisting to Detect Phishing Attacks," in *Proceedings of the 29th Conference on Computer Communications (INFOCOM)*, March 2010.

[186] PROJECT, T. S., "Zen - spamhause dnsbls," 2004.

[187] PROVOS, N., MAVROMMATIS, P., RAJAB, M. A., and MONROSE, F., "All Your iFRAMEs Point to Us," in *Proceedings of the 17th USENIX Security Symposium (SECURITY)*, 2008.

[188] PROVOS, N., MCNAMEE, D., MAVROMMATIS, P., WANG, K., and MODADUGU, N., "The Ghost in the Browser: Analysis of Web-Based Malware," in *Proceedings of the 2nd USENIX Workshop on Hot Topics on Understanding Botnets (HotSec)*, 2007.

[189] RAHBARINIA, B., BALDUZZI, M., and PERDISCI, R., "Real-Time Detection of Malware Downloads via Large-Scale URL→File→Machine Graph Mining," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIACCS)*, 2016.

[190] RAHBARINIA, B., PERDISCI, R., and ANTONAKAKIS, M., "Segugio: Efficient Behavior-Based Tracking of Malware-Control Domains in Large ISP Networks," in *Proceedings of the 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '15, (Washington, DC, USA), pp. 403–414, IEEE Computer Society, 2015.

[191] RAJAB, M. A., BALLARD, L., LUTZ, N., MAVROMMATIS, P., and PROVOS, N., "CAMP: Content-Agnostic Malware Protection," in *Proceedings of the 17th Network and Distributed System Security Symposium (NDSS)*, 2013.

[192] RAJAB, M. A., BALLARD, L., MAVROMMATIS, P., PROVOS, N., and ZHAO, X., "The Nocebo Effect on the Web: An Analysis of Fake Anti-virus Distribution," in *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More (LEET)*, April 2010.

[193] RAMACHANDRAN, A., DAGON, D., and FEAMSTER, N., "Can DNS-based Blacklists Keep Up with Bots?," in *Proceedings of the 3rd Conference on E-mail and Anti-spam (CEAS)*, July 2006.

[194] RAMACHANDRAN, A., FEAMSTER, N., and DAGON, D., "Revealing Botnet Membership Using DNSBL Counter-intelligence," in *Proceedings of the 2nd Conference on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, July 2006.

[195] RAMACHANDRAN, A., FEAMSTER, N., and VEMPALA, S., "Filtering Spam with Behavioral Blacklisting," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*, October 2007.

[196] REKHTER, Y. and LI, T., "An Architecture for IP Address Allocation with CIDR." RFC 1518 (Historic), Sept. 1993.

[197] RICCIATO, F., "Unwanted traffic in 3g networks," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 53–56, 2006.

[198] ROSSOW, C., DIETRICH, C., and BOS, H., "Large-scale Analysis of Malware Downloaders," in *Proceedings of the 9th Conference on Detection of Intrusions and Malware, and Vulernability Assessment (DIMVA)*, 2012.

[199] ROYAL, P., "Analysis of the Kraken Botnet," tech. rep., Damballa Labs, 2008.

[200] SATO, K., ISHIBASHI, K., TOYONO, T., and MIYAKE, N., "Extending Black Domain Name List by Using Co-occurrence Relation Between DNS Queries," in *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More (LEET)*, April 2010.

[201] SCHLAMP, J., GUSTAFSSON, J., WÄHLISCH, M., SCHMIDT, T. C., and CARLE, G., "The Abandoned Side of the Internet: Hijacking Internet Resources When Domain Names Expire," tech. rep., Technische Universität München, Freie Universität Berlin, HAW Hamburg, December 2014.

[202] SCHUMACHER, M., "gimp.org domain has been renewed, DNS updates are still happening," August 2015.

[203] SEBASTIÁN, M., RIVERA, R., KOTZIAS, P., and CABALLERO, J., "AVClass: A Tool for Massive Malware Labeling," in *Proceedings of the 19th International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, 2016.

[204] SEGARAN, T., *Programming Collective Intelligence*. O'Reilly Media, 1st edition ed., 2007.

[205] SHEVCHENKO, S., "Srizbi's Domain Calculator." `http://blog.threatexpert.com/2008/11/srizbis-domain-calculator.html`, 2008.

[206] SHUE, C., KALAFUT, A. J., and GUPTA, M., "Abnormally Malicious Autonomous Systems and Their Internet Connectivity," *IEEE/ACM Transactions of Networking*, 2012.

[207] SORREL, C., "Jailbreakme 3.0: Unlock your ipad 2 from the browser." `https://www.wired.com/2011/07/jailbreakme-3-0-unlock-your-ipad-2-from-the-browser/`, 7 2011.

[208] SPAMHAUS, "DBL: The Domain Block List." `http://www.spamhaus.org/dbl/`, 2015.

[209] SPAMHAUS, "SBL Advistory." `http://www.spamhaus.org/sbl/listings/RIPE`, 2015.

[210] STANIFORD, S., PAXSON, V., and WEAVER, N., "How to 0wn the internet in your spare time," in *Proceedings of the 11th USENIX Security Symposium*, 2002.

[211] STONE-GROSS, B., CHRISTOPHER, KRUEGEL, ALMEROTH, K., MOSER, A., and KIRDA, E., "FIRE: FInding Rogue Networks," in *Proceedings of the 25th Annual Compuer Security Applications Conference (ACSAC)*, 2009.

[212] STROHMEYER, R., "8 notorious Android malware attacks," 8 2011.

[213] SYMANTEC, "W32.Virut." `https://www.symantec.com/security_response/writeup.jsp?docid=2007-041117-2623-99`, 2012.

[214] SZURDI, J., KOCSO, B., CSEH, G., SPRING, J., FELEGYHAZI, M., and KANICH, C., "The Long "Taile" of Typosquatting Domain Names," in *Proceedings of the 23rd USENIX Conference on Security (USENIX Security)*, August 2014.

[215] TAKASHI KATSUKI, "Android.Walkinwat." `http://www.symantec.com/security\_response/writeup.jsp?docid=2011-033008-4831-99`, Mar. 2011.

[216] TAYLOR, T., HU, X., WANG, T., JANG, J., STOECKLIN, M. P., MONROSE, F., and SAILER, R., "Detecting Malicious Exploit Kits Using Tree-based Similarity Searches," in *Proceedings of the 6th ACM Conference on Data and Application Security and Privacy (CODASPY)*, 2016.

[217] TEAM, D. P., "Remove unofficial debian-multimedia.org repository from your sources." `https://bits.debian.org/2013/06/remove-debian-multimedia.html`, 2013.

[218] TEAM CYMRU, I., "Internet security research and insight - team cymru," 2011.

[219] THOMAS, K., GRIER, C., MA, J., PAXSON, V., and SONG, D., "Design and Evaluation of a Real-Time URL Spam Filtering Service," in *Proceedings of the IEEE Computer Society 2011 Security and Privacy Symposium (IEEE S&P)*, May 2011.

[220] THOMAS, K., BURSZTEIN, E., GRIER, C., HO, G., JAGPAL, N., KAPRAVELOS, A., MCCOY, D., NAPPA, A., PAXSON, V., PEARCE, P., PROVOS, N., and RAJAB, M. A., "Ad Injection at Scale: Assessing Deceptive Advertisement Modifications," in *Proceedings of the 36th IEEE Symposium on Security and Privacy (OAKLAND)*, 2015.

[221] THOMAS, M. and MOHAISEN, A., "Kindred Domains: Detecting and Clustering Botnet Domains Using DNS Traffic," in *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, (New York, NY, USA), pp. 707–712, ACM, 2014.

[222] THOMASS, K., CRESPO, J. A. E., RASTIL, R., PICODI, J.-M., BALLARD, L., RAJAB, M. A., PROVOS, N., BURSZTEIN, E., and MCCOY, D., "Investigating Commercial Pay-Per-Install and the Distribution of Unwanted Software," in *Proceedings of the 25th USENIX Security Symposium (SECURITY)*, 2016.

[223] THOMSON, S., HUITEMA, C., KSINANT, V., and SOUISSI, M., "DNS Extensions to Support IP Version 6." RFC 3596 (Draft Standard), Oct. 2003.

[224] TIM WYATT, "DroidDreamLight, new malware from the developers of DroidDream." `http://blog.mylookout.com/blog/2011/05/30/security-alert-droiddreamlight-new-malware-from-the-developers-of-droiddream/`, May 2011.

[225] TRACKER, Z., "Zeus ip and domain name block list.," 2009.

[226] TRAYNOR, P., ENCK, W., MCDANIEL, P., and PORTA, T. L., "Exploiting open functionality in sms-capable cellular networks," *Journal of Computer Security*, vol. 16, no. 6, pp. 713–742, 2008.

[227] TRAYNOR, P., ENCK, W., MCDANIEL, P., and PORTA, T. L., "Mitigating attacks on open functionality in sms-capable cellular networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, pp. 40–53, 2009.

[228] TRAYNOR, P., LIN, M., ONGTANG, M., RAO, V., JAEGER, T., PORTA, T. L., and MCDANIEL, P., "On cellular botnets: Measuring the impact of malicious devices on a cellular network core," in *Proceedings of the ACM Conference on Computer and Communications Security*, 2009.

[229] TRAYNOR, P., MCDANIEL, P., and PORTA, T. L., "On attack causality in internet-connected cellular networks," in *Proceedings of the 16th Network and Distributed System Security Symposium*, 2007.

[230] TRAYNOR, P., AMRUTKAR, C., RAO, V., JAEGER, T., MCDANIEL, P., and LA PORTA, T., "From Mobile Phones to Responsible Devices," *Journal of Security and Communication Networks (SCN)*, vol. 4, pp. 719 – 726, June 2011.

[231] URIBL, "Real time uri blacklist.," 2011.

[232] VIXIE, P., "DNS Complexity," *Queue*, vol. 5, pp. 24–29, Apr. 2007.

[233] WANG, L., NAPPA, A., CABALLERO, J., RISTENPART, T., and AKELLA, A., "WhoWas: A Platform for Measuring Web Deployments on IaaS Clouds," in *Proceedings of the 2014 ACM Internet Measurement Conference (IMC)*, 2014.

[234] WANG, Y.-M., BECK, D., JIANG, X., ROUSSEV, R., VERBOWSKI, C., CHEN, S., and KING, S., "Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities," in *Proceedings of the 10th Network and Distributed System Security Symposium (NDSS)*, 2006.

[235] WEIMER, F., "Passive DNS replication," in *Proceedings of the 1st Conference on Computer Security Incident*, 2005.

[236] WEIMER, F., "Passive DNS Replication," in *In Proceedings of the 17th FIRST Conference on Computer Security Incident Handling*, June 2005.

[237] WESSELS, D., FOMENKOV, M., BROWNLEE, N., and CLAFFY, K., "Measurements and Laboratory Simulations of the Upper DNS Hierarchy," in *Proceedings of the 5th International Passive and Active Measurement Workshop (PAM)*, 2004.

[238] WHITNEY, L., "Android's popularity makes it open target for malware, says study," 12 2011.

[239] WIKIPEDIA, "Mydoom." `https://en.wikipedia.org/wiki/Mydoom`, 2016.

[240] WIKIPEDIA, "Blackberry world." https://en.wikipedia.org/wiki/BlackBerry_World, 2017.

[241] WIKIPEDIA, "Domain name speculation." https://en.wikipedia.org/wiki/Domain_name_speculation, 2017.

[242] WIKIPEDIA, "Package manager." https://en.wikipedia.org/wiki/Package_manager, 2017.

[243] WIKIPEDIA, "Windows phone store." https://en.wikipedia.org/wiki/Windows_Phone_Store, 2017.

[244] WILLEMS, C., HOLZ, T., and FREILING, F., "Toward Automated Dynamic Malware Analysis Using CWSandbox," *IEEE Security & Privacy*, 2007.

[245] XU, W., SANDERS, K., and ZHANG, Y., "We Know It Before You Do: Predicting Malicious Domains," tech. rep., Palo Alto Networks, 2014.

[246] XU, Z., NAPPA, A., BAYKOV, R., YANG, G., CABALLERO, J., and GU, G., "AutoProbe: Towards Automatic Active Malicious Server Probing Using Dynamic Binary Analysis," in *Proceedings of the 21st ACM Conference on Computer and Communication Security (CCS)*, 2014.

[247] YADAV, S., REDDY, A. K. K., REDDY, A. N., and RANJAN, S., "Detecting Algorithmically Generated Malicious Domain Names," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC)*, November 2010.

[248] YEN, T.-F. and REITER, M. K., "Traffic Aggregation for Malware Detection," in *Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2008.

[249] YI LI, "Android.Notcompatible." http://www.symantec.com/security_response/writeup.jsp?docid=2012-050307-2712-99, May 2012.

[250] ZEMAN, E., "Android Security: Threat Level None?," 11 2011.

[251] ZHANG, J., PERDISCI, R., LEE, W., SARFRAZ, U., and LUO, X., "Detecting Stealthy P2P Botnets Using Statistical Traffic Fingerprints," in *Proceedings of the 41st International Conference on Dependable Systems and Networks (DSN)*, 2011.

[252] ZHOU, Y. and JIANG, X., "Dissecting Android Malware: Characterization and Evolution," in *Proceedings of the IEEE Symposium on Security and Privacy (OAKLAND)*, 2012.

[253] ZHOU, Y., WANG, Z., ZHOU, W., and JIANG, X., "Hey, You, Get off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets," in *Proceedings of the 19th Network and Distributed System Security Symposium (NDSS)*, 2012.

[254] ZHOU, Y., WANG, Z., ZHOU, W., and JIANG, X., "Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets," in *Proceedings of the 19th Network and Distributed System Security Symposium (NDSS)*, 2012.